Journal of Artificial Intelligence Research 30 (2007) 413-456 Submitted 05/2007; published 11/2007

Individual and Domain Adaptation in Sentence Planning for Dialogue

Marilyn Walker LYNWALKER@GMAIL.COM
Department of Computer Science, University of Sheffield
211 Portobello Street, Sheffield S1 4DP, United Kingdom

Amanda Stent AMANDA.STENT@GMAIL.COM
Department of Computer Science, Stony Brook University
Stony Brook, NY 11794, USA

François Mairesse F.MAIRESSE@SHEFFIELD.AC.UK
Department of Computer Science, University of Sheffield,
211 Portobello Street, Sheffield S1 4DP, United Kingdom

Rashmi Prasad RJPRASAD@LINC.CIS.UPENN.EDU
Institute for Research in Cognitive Science, University of Pennsylvania,
3401 Walnut Street, Suite 400A, Philadelphia, PA 19104, USA

Abstract

One of the biggest challenges in the development and deployment of spoken dialogue systems is the design of the spoken language generation module. This challenge arises from the need for the generator to adapt to many features of the dialogue domain, user population, and dialogue context. A promising approach is trainable generation, which uses general-purpose linguistic knowledge that is automatically adapted to the features of interest, such as the application domain, individual user, or user group. In this paper we present and evaluate a trainable sentence planner for providing restaurant information in the MATCH dialogue system. We show that trainable sentence planning can produce complex information presentations whose quality is comparable to the output of a template-based generator tuned to this domain. We also show that our method easily supports adapting the sentence planner to individuals, and that the individualized sentence planners generally perform better than models trained and tested on a population of individuals. Previous work has documented and utilized individual preferences for content selection, but to our knowledge, these results provide the first demonstration of individual preferences for sentence planning operations, affecting the content order, discourse structure and sentence structure of system responses. Finally, we evaluate the contribution of different feature sets, and show that, in our application, n-gram features often do as well as features based on higher-level linguistic representations.

1. Introduction

One of the most robust findings of studies of human-human dialogue is that people adapt their interactions to match their conversational partners' needs and behaviors (Goffman, 1981; Brown & Levinson, 1987; Pennebaker & King, 1999). People adapt the content of their utterances (Garrod & Anderson, 1987; Luchok & McCroskey, 1978). They choose syntactic structures to match their partners' syntax (Levelt & Kelter, 1982; Branigan, Pickering, & Cleland, 2000; Reitter, Keller, & Moore, 2006; Stenchikova & Stent, 2007), and adapt their choice of words and referring expressions (Clark & Wilkes-Gibbs, 1986; Brennan & Clark, 1996). They also adapt their speaking rate, amplitude, and clarity of pronunciation (Jungers, Palmer, & Speer, 2002; Coulston, Oviatt, & Darves, 2002; Ferguson & Kewley-Port, 2002).

However, it is beyond the state of the art to reproduce this type of adaptation in the *spoken language generation module* of a dialogue system, i.e. the components that handle response generation and information presentation. A standard generation system includes modules for content planning, sentence planning, and surface realization (Kittredge, Korelsky, & Rambow, 1991; Reiter & Dale, 2000). A **content planner** takes as input a communicative goal; it selects content to realize that goal and organizes that content into a *content plan*. A **sentence planner** takes as input a content plan. It decides how the content is allocated into sentences, how the sentences are ordered, and which discourse cues to use to express the relationships between content elements. It outputs a *sentence plan*. Finally, a **surface realizer** determines the words and word order for each sentence in the sentence plan. It outputs a text or speech *realization* for the original communicative goal.

The findings from human-human dialogue suggest that adaptation could potentially be useful at any stage of the generation pipeline. Yet to date, the only work on adaptation to individual users utilizes models of the user's knowledge, needs, or preferences to adapt the *content* for content planning (Jokinen & Kanto, 2004; Rich, 1979; Wahlster & Kobsa, 1989; Zukerman & Litman, 2001; Carenini & Moore, 2006), rather than applying models of individual linguistic preferences as to the *form* of the output, as determined by sentence planning or surface realization.

However, consider the alternative realizations for a restaurant recommendation in Figure 1. Columns A and B contain human ratings of the quality of the realizations from users A and B. The differences in the rating feedback suggest that each user has different perceptions as to the quality of the potential realizations. Data from an experiment collecting feedback from users A and B, for 20 realizations of 30 different recommendation content plans (600 examples), shows that the feedback of the two users are easily distinguished: a paired t-test supports the hypothesis that the two samples are sampled from distinct distributions ($t = 17.4$, $p < 0.001$). These perceptual differences appear to be more general: when we examined the user feedback from the evaluation experiment described by Rambow, Rogati, and Walker (2001) where 60 users rated the output of 7 different spoken language generators for 20 content plans, we again found significant differences in user perceptions of utterance quality ($F = 1.2$, $p < 0.002$). This led us to hypothesize that individualized sentence planners for dialogue systems might be of high utility.

In addition to our own studies, we also find evidence in other work that individual variation is inherent to many aspects of language generation, including content ordering, referring expression generation, syntactic choice, lexical choice, and prosody generation.

- It is common knowledge that individual authors can be identified from the linguistic features of their written texts (Madigan, Genkin, Lewis, Argamon, Fradkin, & Ye, 2005; Oberlander & Brew, 2000).
- An examination of a weather report corpus for five weather forecasters showed individual differences in lexical choice for expressing specific weather-related concepts (Reiter & Sripada, 2002).

Alt	Realization	A	B	AVG
6	Chanpen Thai has the best overall quality among the selected restaurants since it is a Thai restaurant, with good service, its price is 24 dollars, and it has good food quality.	1	4	2.5
7	Chanpen Thai has the best overall quality among the selected restaurants because it has good service, it has good food quality, it is a Thai restaurant, and its price is 24 dollars.	2	5	3.5
4	Chanpen Thai has the best overall quality among the selected restaurants. It has good food quality, with good service, it is a Thai restaurant, and its price is 24 dollars.	2	4	3
9	Chanpen Thai is a Thai restaurant, with good food quality, its price is 24 dollars, and it has good service. It has the best overall quality among the selected restaurants.	2	4	3
5	Chanpen Thai has the best overall quality among the selected restaurants. It has good service. It has good food quality. Its price is 24 dollars, and it is a Thai restaurant.	3	2	2.5
3	Chanpen Thai has the best overall quality among the selected restaurants. Its price is 24 dollars. It is a Thai restaurant, with good service. It has good food quality.	3	3	3
10	Chanpen Thai has the best overall quality among the selected restaurants. It has good food quality. Its price is 24 dollars. It is a Thai restaurant, with good service.	3	3	3
2	Chanpen Thai has the best overall quality among the selected restaurants. Its price is 24 dollars, and it is a Thai restaurant. It has good food quality and good service.	4	4	4
1	Chanpen Thai has the best overall quality among the selected restaurants. This Thai restaurant has good food quality. Its price is 24 dollars, and it has good service.	4	3	3.5
8	Chanpen Thai is a Thai restaurant, with good food quality. It has good service. Its price is 24 dollars. It has the best overall quality among the selected restaurants.	4	2	3

Figure 1: Some alternative realizations for the content plan in Figure 4, with feedback from Users A and B, and the mean (AVG) of their feedback (1=worst and 5=best).

- Rules learned for generating nominal referring expressions perform better when individual speakers are provided as a feature to the learning algorithm (Jordan & Walker, 2005), and an experiment evaluating choice of referring expression shows only 70% agreement among native speakers as to the best choice (Yeh & Mellish, 1997). Chai, Hong, Zhou, and Prasov (2004) show that there are also individual differences in gesture when generating multimodal references, and the corpus study of accented pronouns reported by Kothari (2007) suggests that accentuation is also partly determined by individual linguistic style.

- Automatic evaluation techniques applied to human-generated reference outputs for machine translation and automatic summarization perform better when multiple outputs are provided for comparison (Papenini, Roukos, Ward, & Zhu, 2002; Nenkova, Passonneau, & McKeown, 2007): this can be attributed to the large variation in what humans generate given particular content to express. This is also reflected in the finding that human subjects produce many different valid content orderings when asked to order a specific set of content items to produce the best possible summary (Barzilay, Elhadad, & McKeown, 2002; Lapata, 2003).

In the past, linguistic variation among individuals was considered a *problem* for generation researchers to work around, rather than a potential area of study (McKeown, Kukich, & Shaw, 1994; Reiter, 2002; Reiter, Sripada, & Robertson, 2003). In part, this was due to the hand-crafting of generation components and resources. It is impossible to encode by hand, for each individual, rules for sentence planning and realization. Furthermore, if domain experts don't agree on the best way to express a domain concept, how can the generation dictionary be encoded? It is difficult simply to get good output that respects all the interacting domain and linguistic constraints even with considerable handcrafting of rules (Kittredge, Korelsky, & Rambow, 1991).

Modeling individual differences can also be a problem for statistical methods when learning paradigms are used that assume there is a single correct output (Lapata, 2003; Jordan & Walker, 2005; Hardt & Rambow, 2001) *inter alia*. We believe that the simplest way to deal with the inherent variability in possible generation outputs is to treat generation as a ranking problem as we explain below, with techniques that overgenerate using user or domain-independent rules, and then filter or rank the possibilities using domain or user-specific corpora or feedback (Langkilde & Knight, 1998; Langkilde-Geary, 2002; Bangalore & Rambow, 2000; Rambow, Rogati, & Walker, 2001). This approach has an advantage for dialogue systems because it also affords joint optimization of the generator and the text-to-speech engine (Bulyko & Ostendorf, 2001; Nakatsu & White, 2006). There are many problems in generation to which ranking models and individualization could be applied, such as text planning, cue word selection, or referring expression generation (Mellish, O'Donnell, Oberlander, & Knott, 1998; Litman, 1996; Di Eugenio, Moore, & Paolucci, 1997; Marciniak & Strube, 2004). However, only recently has any work in generation acknowledged that there are individual differences and tried to model them (Guo & Stent, 2005; Mairesse & Walker, 2005; Belz, 2005; Lin, 2006).

This article describes SPaRKy (Sentence Planning with Rhetorical Knowledge), a sentence planner that uses rhetorical relations and adapts to the user's individual sentence planning preferences.[1] SPaRKy has two components: a randomized sentence plan generator (SPG) that produces multiple alternative realizations of an information presentation, and a sentence plan ranker (SPR) that is trained (using human feedback) to rank these alternative realizations (See Figure 1). As mentioned above, previous work has documented and utilized individual preferences for content selection, but to our knowledge, our results provide the first demonstration of individual preferences for sentence planning operations, affecting the content ordering, discourse structure, sentence structure, and sentence scope of system responses. We also show that some of the learned preferences are domain-specific.

Section 2 compares our approach and results with previous work. Section 3 provides an overview of the MATCH system architecture, which can generate dialogue system responses using either SPaRKy, or a domain-specific template-based generator described and evaluated in previous work (Stent, Walker, Whittaker, & Maloor, 2002; Walker et al., 2004). Sections 4, 5 and 6 describe SPaRKy in detail; they describe the SPG, the automatic generation of features used in training the SPR, and how boosting is used to train the SPR. Sections 7 and 8 present both quantitative and qualitative results:

1. A Java version of SPaRKy can be downloaded from `www.dcs.shef.ac.uk/cogsys/sparky.html`

1. First, we show that SPaRKy learns to select sentence plans that are significantly better than a randomly selected sentence plan, and on average less than 10% worse than a sentence plan ranked highest by human judges. We also show that, in our experiments, simple n-gram features perform as well as features based on higher-level linguistic representations.

2. Second, we show that SPaRKy's SPG can produce realizations that are comparable to that of MATCH's template-based generator, but that there is a gap between the realization that the SPR selects when trained on multiple users and those selected by a human.

3. Third, we show that when SPaRKy is trained for particular individuals, it performs better than when trained on feedback from multiple individuals. These are the first results suggesting that individual sentence planning preferences exist, and that they can be modeled by a trainable generation system. We also show that in most cases the performance of the individualized SPRs are statistically indistinguishable from MATCH's template-based generator, but for compare-2, User B prefers SPaRKy, while for compare-3, User A prefers the template-based generator.

4. Fourth, we show that the differences in the learned models make sense in terms of previous rule-based approaches to sentence planning. We analyze the qualitative differences between the learned group and individual models, and show that SPaRKy learns specific rules about the interaction between content items and sentence planning operations, and rules that model individual differences, that would be difficult to capture with a hand-crafted generator.

We sum up and discuss future work in Section 9.

2. Related Work

We discuss related work on adaptation in generation using the standard generation architecture which contains modules for content planning (Section 2.1), sentence planning (Section 2.2) and surface realization (Section 2.3) (Kittredge, Korelsky, & Rambow, 1991; Reiter & Dale, 2000).

2.1 Adaptation in Content Planning

There has been significant research on the use of user models and discourse context to adapt the content of information presentations in dialogue (Joshi, Webber, & Weischedel, 1984, 1986; Chu-Carroll & Carberry, 1995; Zukerman & Litman, 2001) *inter alia*, but only the user models (not the information presentation strategies) are sensitive to particular individuals. Several studies have investigated the use of quantitative models of user preferences in selection of content for recommendations and comparisons (Carenini & Moore, 2006; Walker et al., 2004; Polifroni & Walker, 2006), and Moore, Foster, Lemon, and White (2004) use such models for referring expression generation, sentence planning and some surface realization. Elhadad, Kan, Klavans, and McKeown (2005) applied group models (physician, lay person) and individual user models to the task of summarizing medical information.

McCoy (1989) used context information to design helpful system-generated corrections. Other work has looked at the use of statistical techniques for adapting content selection and content ordering methods to particular domains (Barzilay, Elhadad, & McKeown, 2002; Duboue & McKeown, 2003; Lapata, 2003), but not to individual users.

2.2 Adaptation in Sentence Planning

The first trainable sentence planner was SPoT, a precursor to SPaRKy that output information gathering utterances in the travel domain (Walker, Rambow, & Rogati, 2002). Evaluations of SPoT demonstrated that it performed as well as a template-based generator developed for the travel domain and field-tested in the DARPA Communicator evaluations (Rambow, Rogati, & Walker, 2001; Walker et al., 2002). Information gathering utterances are considerably simpler than information presentations: they do not usually exhibit any complexities in rhetorical structure, and there is little interaction between domain-specific content items and sentence structures. Thus the SPoT generator did not produce utterances with variation in rhetorical structure; it learned to optimize speech-act ordering and sentence structure choices, but it did not adapt to individuals.

2.3 Adaptation in Surface Realization

Work on adaptation in surface realization has mainly focused on decisions such as lexical and syntactic choice, using models of a target text, but not individual text models, although recent research has also shown that n-gram models trained on user-specific corpora can adapt generators to reproduce individualized lexical and syntactic choices (Lin, 2006; Belz, 2005). Paiva and Evans (2004) present a technique for training a generator by learning the relationship between particular generation decisions and text variables that can be measured in the output corpus. This technique was applied to generator decisions such as the form of referring expression and syntactic structure, and was used to capture stylistic, rather than individual, differences. Gupta and Stent (2005) use discourse context and speaker knowledge for referring expression generation in dialogue.

User models have also been used to adapt surface realization. The approach of learning a ranking from user feedback has been applied to multimedia presentation planning (Stent & Guo, 2005) and to the joint optimization of the syntactic realizer and the text-to-speech engine (Nakatsu & White, 2006). This work does not look at individual differences.

Research has also focused on other factors that affect stylistic variation – how realization choices reflect personality, politeness, emotion or domain specific style (Hovy, 1987; DiMarco & Foster, 1997; Walker, Cahn, & Whittaker, 1997; André, Rist, van Mulken, Klesen, & Baldes, 2000; Bouayad-Agha, Scott, & Power, 2000; Fleischman & Hovy, 2002; Piwek, 2003; Porayska-Pomsta & Mellish, 2004; Isard, Brockmann, & Oberlander, 2006; Gupta, Walker, & Romano, 2007; Mairesse & Walker, 2007). None of this work has attempted to reproduce individual stylistic variation.

3. Overview of MATCH's Spoken Language Generator

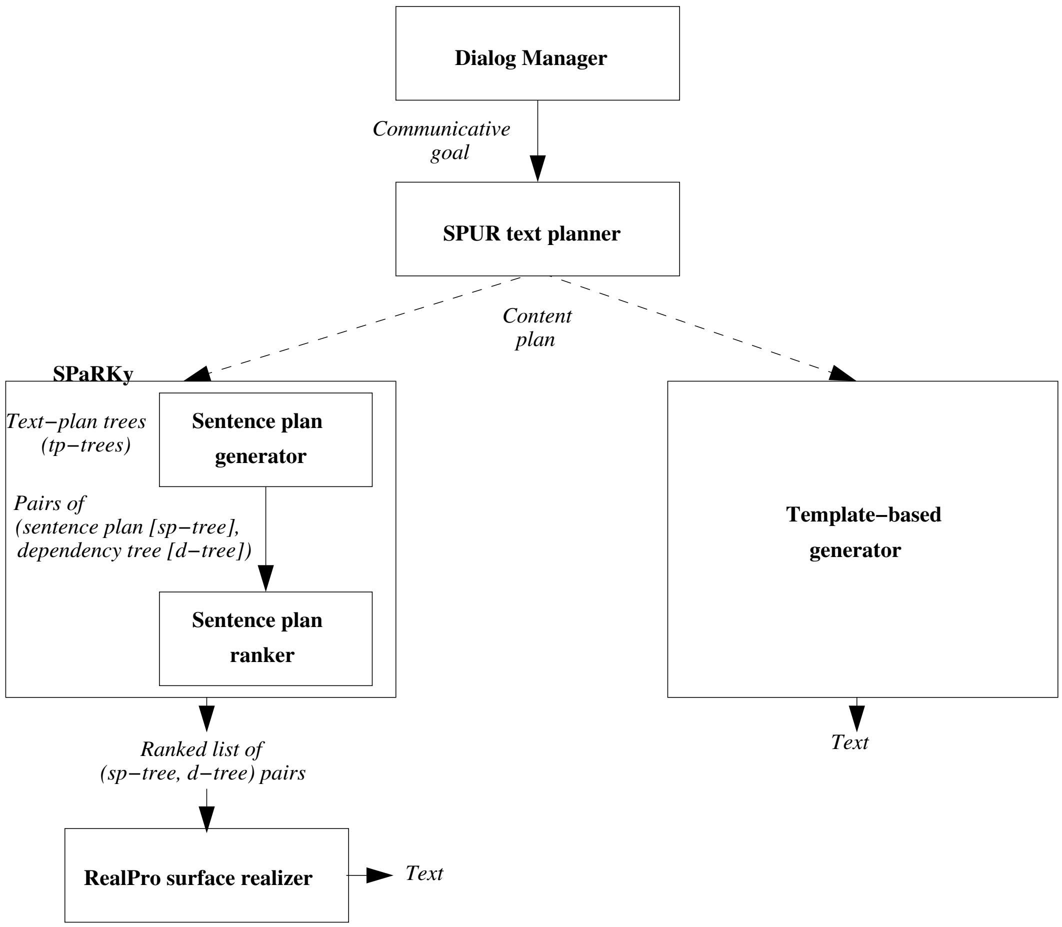

Figure 2: Architecture of MATCH's Spoken Language Generator.

MATCH (Multimodal Access To City Help) is a multimodal dialogue system for finding restaurants and entertainment options in New York City (Johnston, Bangalore, Vasireddy, Stent, Ehlen, Walker, Whittaker, & Maloor, 2002). Information presentations in MATCH include route descriptions, as well as user-tailored *recommendations* and *comparisons* of restaurants. Figure 2 shows MATCH's architecture for spoken language generation (SLG). The content planning module is the SPUR text planner (Section 3.1) (Walker et al., 2004). There are two modules for producing text or spoken dialogue responses from SPUR's output: a highly engineered domain-specific template-based realizer (Section 3.2); and the SPaRKy sentence planner followed by the RealPro surface realizer (Lavoie & Rambow, 1997) (Section 3.3). Example template-based and SPaRKy outputs for each dialogue strategy are in Figure 3. Both SPUR and SPaRKy are trainable, and produce different output depending on the user and discourse context.

Strategy	System	Realization	AVG
RECOMMEND	Template	Caffe Cielo has the best overall value among the selected restaurants. Caffe Cielo has good decor and good service. It's an Italian restaurant.	4
RECOMMEND	SPaRKy	Caffe Cielo, which is an Italian restaurant, with good decor and good service, has the best overall quality among the selected restaurants.	4
COMPARE-2	Template	Caffe Buon Gusto's an Italian restaurant. On the other hand, John's Pizzeria's an Italian, Pizza restaurant.	2
COMPARE-2	SPaRKy	Caffe Buon Gusto is an Italian restaurant, and John's Pizzeria is an Italian , Pizza restaurant.	4
COMPARE-3	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
COMPARE-3	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 3: Template outputs and a sample SPaRKy output for each dialogue strategy. AVG = Averaged score of two human users.

3.1 SPUR

The input to SPUR is a high-level **communicative goal** from the MATCH dialogue manager and its output is a **content plan** for a recommendation or comparison. SPUR selects and organizes the content to be communicated based on the communicative goal, a conciseness parameter, and a decision-theoretic user model. It produces *targeted* recommendations and comparisons: the restaurants mentioned and the attributes selected for each restaurant are those the user model predicts the user will want to know about. Thus SPUR can produce a wide variety of content plans.

Figure 4 shows a sample content plan for a recommendation. This content plan gives rise to the alternate realizations for recommendations for Chanpen Thai in Figure 1. Following a bottom-up approach to text-planning (Marcu, 1997; Mellish, O'Donnell, Oberlander, & Knott, 1998), each content plan consists of a set of *assertions* that must be communicated to the user and a set of *rhetorical relations* that hold between those assertions that may be communicated as well. Each rhetorical relation designates one or more facts as the *nuclei* of the relation, i.e. the main point, and the other facts as *satellites*, i.e. the supplementary facts (Mann & Thompson, 1987). Three rhetorical relations (Mann & Thompson, 1987) are used by SPUR: the JUSTIFY relation for the recommendation strategy, and the CONTRAST and ELABORATION relations for the comparison strategies. The relations in Figure 4 specify that the nucleus (1) is the *claim* being made in the recommendation, and that the satellites (assertions 2 to 5) provide justifying evidence for the claim.

```
relations:justify(nuc:1, sat:2); justify (nuc:1, sat:3 ); justify(nuc:1, sat:4);
          justify(nuc:1, sat:5)
content:  1. assert(best (Chanpen Thai))
          2. assert(is (Chanpen Tai, cuisine(Thai)))
          3. assert(has-att(Chanpen Thai, food-quality(good)))
          4. assert(has-att(Chanpen Thai, service(good)))
          5. assert(is (Chanpen Thai, price(24 dollars)))
```

Figure 4: A content plan for a recommendation.

3.2 Template-Based Generator

In order to produce utterances from the content plans produced by SPUR, we first implemented and evaluated a template-based generator for MATCH (Stent, Walker, Whittaker, & Maloor, 2002; Walker et al., 2004). The template-based generator was designed to make it possible to evaluate algorithms for user-specific content selection based on SPUR's decision-theoretic user model. It performs sentence planning, including some discourse cue insertion, clause combining and referring expression generation. It produces one high quality output for any content plan for our three dialogue strategies: RECOMMEND, COMPARE-2 and COMPARE-3. Recommendations and comparisons are one form of *evaluative argument*, so its realization strategies are based on guidelines from argumentation theory for producing effective evaluative arguments, as summarized by Carenini and Moore (2000). Because the templates are highly tailored to this domain, the template-based generator can be expected to perform well in comparison to SPaRKy.

Following the argumentation guidelines, the template-based generator realizes recommendations with the nucleus ordered first, followed by the satellites. The satellites are ordered to maximize the opportunity for aggregation. To produce the most concise recommendations given the content to be communicated, phrases with identical verbs and subjects are grouped, so that lists and coordination can be used to aggregrate the assertions about the subject. Figure 5 provides examples of aggregration as the number of assertions varies according to SPUR's conciseness parameter (Z-value).

The realization template for comparisons focuses on communicating both the *elaboration* and the *contrast* relations. Figure 6 contains a content plan for comparisons. The nucleus is the assertion (1) that Above and Carmine's are exceptional restaurants. The satellites (assertions 2 to 7 representing the selected attributes for each restaurant) *elaborate* on the claim in the nucleus (assertion 1). *Contrast* relations hold between assertions 2 and 3, between 4 and 5, and between 6 and 7. One way to communicate the *elaboration* relation is to structure the comparison so that all the satellites are grouped together, following the nucleus. To communicate the *contrast* relation, the satellites are produced in a fixed order, with a parallel structure maintained across options (Prevost, 1995; Prince, 1985). The satellites are initially ordered in terms of their evidential strength, but then are reordered to allow for aggregation. Figure 7 illustrates aggregation for comparisons with varying numbers of assertions.

Z	Output
1.5	Komodo has the best overall value among the selected restaurants. Komodo's a Japanese, Latin American restaurant.
0.7	Komodo has the best overall value among the selected restaurants. Komodo's a Japanese, Latin American restaurant.
0.3	Komodo has the best overall value among the selected restaurants. Komodo's price is $29. It's a Japanese, Latin American restaurant.
-0.5	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service. It's a Japanese, Latin American restaurant.
-0.7	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service and very good food quality. It's a Japanese, Latin American restaurant.
-1.5	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service, very good food quality and good decor. It's a Japanese, Latin American restaurant.

Figure 5: Recommendations for the East Village Japanese Task, for different settings of the conciseness parameter Z.

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

Figure 6: A content plan for a comparison.

3.3 SPaRKy

Like the template-based generator, SPaRKy takes as input any of the content plans produced by SPUR. Figure 2 shows that SPaRKy has two modules: the sentence plan generator (SPG), and the sentence plan ranker (SPR). The SPG uses a set of clause-combining operations (Figure 12); it produces a large set of alternative realizations of an input content plan (See Figure 1). The SPR ranks the alternative realizations using a model learned from users' ratings of a training set of content plans. The SPG is described in Section 4. The features used to train the SPR are described in Section 5; the procedure for training the SPR is described in Section 6.

Because SPaRKy is trained using user feedback, rather than being handcrafted, it can be trained to be an individualized spoken language generator. As discussed above, the

Z	Output
1.5	Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service.
0.7	Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service and good decor.
0.3	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor.
-0.5	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor. Japonica's price is$37. It has excellent food quality, good service and decent decor
-0.7	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor. Japonica's price is $37. It has excellent food quality, good service and decent decor. Shabu-Tatsu's price is $31. It has very good food quality, good service and decent decor.
-1.5	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor. Japonica's price is $37. It has excellent food quality, good service and decent decor. Shabu-Tatsu's price is $31. It has very good food quality, good service and decent decor. Bond Street's price is $51. It has excellent food quality, good service and very good decor. Dojo's price is $14. It has decent food quality, mediocre service and mediocre decor.

Figure 7: Comparisons for the East Village Japanese Task, for different settings of the conciseness parameter Z.

feedback from the two users in Figure 1 suggests that each user has different perceptions as to the quality of the potential realizations. A significant part of Sections 7 and 8 are dedicated to examining the differences between a model trained on averaged feedback, shown as AVG in Figure 1, and those trained on individual feedback from users A and B.

4. Sentence Plan Generation

The input to SPaRKy's SPG is a **content plan** from SPUR. Content plans for a sample recommendation and comparison were in Figure 4 and Figure 6. Figure 1 shows alternative SPaRKy realizations for the recommendation in Figure 4, while Figure 8 shows alternative SPaRKy realizations for the comparison in Figure 6. Content plans specify which assertions to include in an information presentation, and the rhetorical relations holding between them, but not the order of assertions or how to express the rhetorical relations between them. This task is known as *discourse planning*. The SPG has two stages of processing; first it does discourse planning, and then it does sentence planning.

4.1 Discourse Planning

Discourse planning algorithms can be characterized as: schema-based (McKeown, 1985; Kittredge, Korelsky, & Rambow, 1991); top-down algorithms using plan operators (Moore & Paris, 1993); or bottom-up approaches that use, for example, constraint satisfaction algorithms (Marcu, 1996, 1997) or genetic algorithms (Mellish, O'Donnell, Oberlander, &

Alt	Realization	A	B	AVG
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	2	2
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.	3	2	2.5
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	3	3
14	Above and Carmine's offer exceptional value among the selected restaurants. Above has good decor while Carmine's has decent decor, and Above and Carmine's have good service. Above is a New American restaurant while Carmine's is an Italian restaurant.	4	5	4.5
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	3	2.5
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	NR

Figure 8: Some alternative realizations for the compare-3 plan in Figure 6, with feedback from Users A and B, and the mean (AVG) of their feedback (1=worst and 5=best). NR = Not generated or ranked.

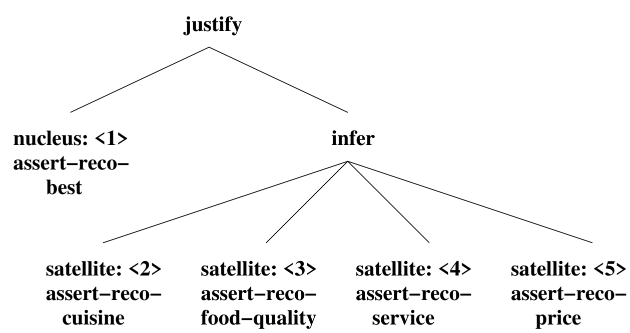

Figure 9: A tp-tree for the plan of Figure 4, used to generate Alternatives 1, 3, 4, 5, 6, 7 and 10 in Figure 1.

Knott, 1998). In SPaRKy, the SPG takes a bottom-up approach to discourse planning using principles from Centering Theory (Grosz, Joshi, & Weinstein, 1995). Content items are grouped because they talk about the same thing, but the linear order between and among the groupings is left unspecified. The centering constraints have the result that Alt-25 in Figure 8, which repeatedly changes the discourse center, are never generated.

The discourse planning stage produces one or more text-plan trees (**tp-trees**). A tp-tree for the recommend plan in Figure 4 is in Figure 9, and tp-trees for the compare-3 plan in Figure 6 are in Figure 10. In a tp-tree, each leaf represents a single assertion and is labeled

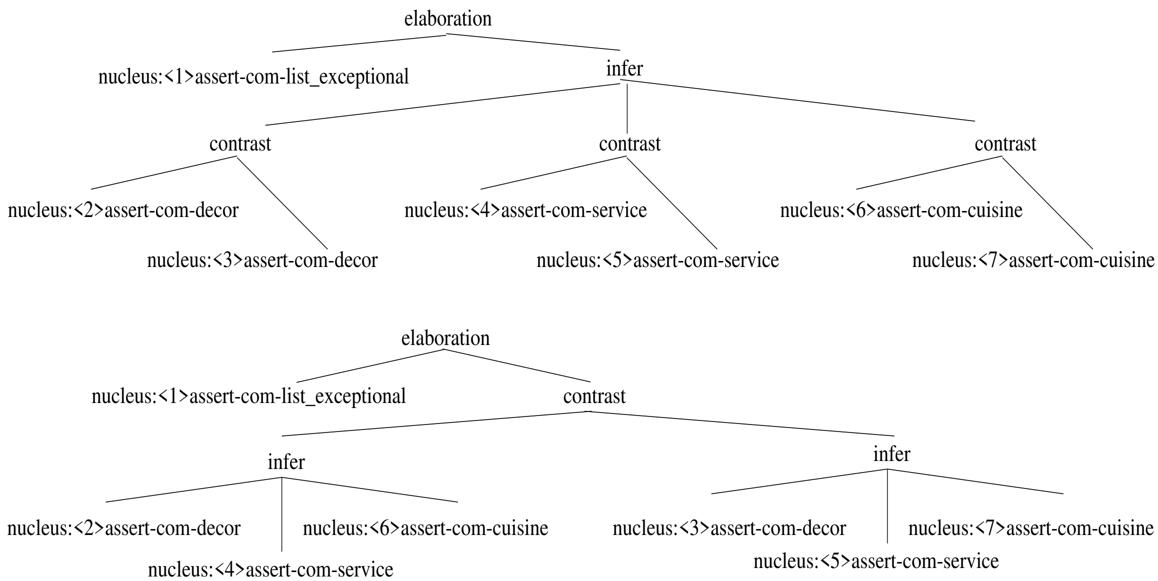

Figure 10: Tp-trees for the comparisons shown as alternatives 12 and 14 (top) and alternatives 11 and 13 (bottom) in Figure 8.

with a speech act. Interior nodes are labeled with rhetorical relations. In addition to the rhetorical relations in the content plan, the SPG uses the relation INFER for combinations of speech acts for which there is no rhetorical relation expressed in the content plan (Marcu, 1997). The *infer* relation is similar to the *joint* relation in RST; it joins multiple satellites in a mononuclear relation or the nuclei in a multinuclear relation.

Each simple assertion, or leaf, in a tp-tree is associated with one or more syntactic realizations (**d-trees**), using a dependency tree representation, called DSyntS (Figure 11) (Melčuk, 1988; Lavoie & Rambow, 1997). The association between the simple assertions and any potential d-trees specifying their syntactic realizations is specified in a hand-crafted generation dictionary. Leaves of some d-trees in the generation dictionary are variables, which are instantiated from the content plan, e.g. *Thai* replaces a cuisine type variable.

4.2 Sentence Planning

During sentence planning, the SPG assigns assertions to sentences, orders the sentences, inserts discourse cues, and performs referring expression generation. It uses a set of clause-combining operations that operate on tp-trees and incrementally transform the elementary d-trees associated with their leaves into a single lexico-structural representation. The output from this process is two parallel structures: (1) a sentence plan tree (**sp-tree**), a binary tree with leaves labeled with the assertions from the input tp-tree, and interior nodes labeled with clause-combining operations; and (2) one or more **d-trees** which reflect parallel operations on the predicate-argument representations.

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

assert-com-cuisine	BE3 [class:verb] (I Chanpen_Thai [number:sg class:proper_noun article:no-art person:3rd] II restaurant [class:common_noun article:indef] (Thai [class:adjective]))
assert-com-food_quality	HAVE1 [class:verb] (I Chanpen_Thai [number:sg class:proper_noun article:no-art person:3rd] II quality [class:common_noun article:no-art] (ATTR good [class:adjective] ATTR food [class:common_noun]))

Figure 11: Example d-trees from the generation dictionary used by the SPG.

examples of their use are given in Figure 12. They are applied in a bottom-up left-to-right fashion, with the choice of operation constrained by the rhetorical relation holding between the assertions to be combined (Scott & de Souza, 1990), as specified in Figure 12.

In addition to ordering assertions, a clause-combining operation may insert cue words between assertions. Figure 13 gives the list of cue words used by the SPG. The choice of cue-word is determined by the type of rhetorical relation[2].

The SPG generates a random sample of possible sp-trees for each tp-tree, up to a pre-specified number of sp-trees, by randomly selecting among the clause-combining operations according to a probability distribution that favors preferred operations. Table 14 shows the probability distribution used in our experiments, which is hand-crafted based on assumed preferences for operations such as MERGE, RELATIVE-CLAUSE and WITH-REDUCTION, and is one way in which some knowledge can be injected into the random process to bias it towards producing higher quality sentence plans.[3]

The SPG handles referring expression generation by converting a proper name to a pronoun when the same proper name appears in the previous utterance. Referring expression generation rules are applied locally, across adjacent utterances, rather than globally across the entire presentation at once (Brennan, Friedman, & Pollard, 1987). Referring expressions are manipulated in the d-trees, either intrasententially during the incremental creation of the sp-tree, or intersententially, if the full sp-tree contains any PERIOD operations. The

2. An alternative approach is for the cue-word to impose a constraint on the rhetorical relation that must hold (Webber, Knott, Stone, & Joshi, 1999; Forbes, Miltsakaki, Prasad, Sarkar, Joshi, & Webber, 2003).
3. This probability distribution could be learned from a corpus (Marcu, 1997; Prasad, Joshi, Dinesh, Lee, & Miltsakaki, 2005).
4. If an INFER relation holds and both clauses contain the HAVE possession predicate, the second clause is arbitrarily selected for reduction. If a JUSTIFY relation holds, it is the satellite of the RST relation that always undergoes reduction, if the syntactic constraints are satisfied.
5. If an INFER relation holds, any clause is arbitrarily selected for reduction. If a JUSTIFY relation holds, the clause that undergoes relative clause formation is the satellite clause. This is motivated by the fact that relative clause formation is generally seen to occur when the modifying relative clause provides additional information about the noun it modifies, but where the additional/elaborated information does not have the same *informational* status as the information in the main clause.

Operation	Rel	Description	Sample 1st arg	Sample 2nd arg	Result
Merge	infer or contrast	Two clauses can be combined if they have identical matrix verbs and identical arguments and adjuncts except one. The non-identical arguments are coordinated.	Chanpen Thai has good service.	Chanpen Thai has good food quality.	Chanpen Thai has good service and good food quality.
With-reduction	justify or infer	Two clauses with identical subject arguments can be identified if one of the clauses has a have-possession matrix verb. The possession clause undergoes *with*-participial clause formation and is attached to the non-reduced clause.[4]	Chanpen Thai is a Thai restaurant.	Chanpen Thai has good food quality.	Chanpen Thai is a Thai restaurant, with good food quality.
Relative-clause	justify or infer	Two clauses with an identical subject can be identified. One clause is attached to the subject of the other clause as a relative clause.[5]	Chanpen Thai has the best overall quality among the selected restaurants.	Chanpen Thai is located in Midtown West.	Chanpen Thai, which is located in Midtown West, has the best overall quality among the selected restaurants.
Cue-word-conjunction	justify, infer or contrast	Two clauses are conjoined with a cue word (coordinating or subordinating conjunction). The order of the arguments of the connective is determined by the order of the nucleus (N) and the satellite (S), yielding two distinct operations, cue-word-conjunction-ns and cue-word-conjunction-sn.	Chanpen Thai has the best overall quality among the selected restaurants.	Chanpen Thai is a Thai restaurant, with good service.	Chanpen Thai has the best overall quality among the selected restaurants, since it is a Thai restaurant, with good service.
Cue-word-insertion (*on the other hand*)	contrast	cue-word insertion combines clauses by inserting a cue word at the start of the second clause (*Carmine's is an Italian restaurant. HOWEVER, Above is a New American restaurant*), resulting in two separate sentences.	Penang has very good decor.	Baluchi's has mediocre decor.	Penang has very good decor. On the other hand, Baluchi's has mediocre decor.
Period	justify, contrast, infer or elaboration	Two clauses are joined by a period.	Chanpen Thai is a Thai restaurant, with good food quality.	Chanpen Thai has good service.	Chanpen Thai is a Thai restaurant, with good food quality. It has good service.

Figure 12: Clause combining operations and examples.

third and fourth sentences for Alt 13 in Figure 8 show the conversion of a named restaurant (*Carmine's*) to a pronoun.

The **sp-trees** for Alts 6 and 8 in Figure 1 are shown in Figs. 15 and 16. Leaf labels are concise names for assertions in the content plan, e.g. **assert-reco-best** is the claim (labelled 1) in Figure 4. Because combination operations can switch the order of their arguments, from satellite before nucleus (SN) to nucleus before satellite (NS), the labels on the interior nodes indicate whether this occurred, and specify the rhetorical relation that the operation realizes. These labels keep track of the operations and substitutions used in constructing the tree and are subsequently used in the tree feature set described in Section 5, one of the

RST relation	Aggregation operator
JUSTIFY	WITH-REDUCTION, RELATIVE-CLAUSE, CUE-WORD CONJ. *because*, CUE-WORD CONJ. *since*, PERIOD
CONTRAST	MERGE, CUE-WORD INSERT. *however*, CUE-WORD CONJ. *while*, CUE-WORD CONJ. *and*, CUE-WORD CONJ. *but*, CUE-WORD INSERT. *on the other hand*, PERIOD
INFER	MERGE, CUE-WORD CONJ. *and*, PERIOD
ELABORATION	PERIOD

Figure 13: RST relation constraints on aggregation operators.

Aggregation operator	Probability
MERGE, WITH-REDUCTION, RELATIVE-CLAUSE	0.80
CUE-WORD CONJ. *because*, CUE-WORD CONJ. *since*, CUE-WORD CONJ. *while*, CUE-WORD CONJ. *and*, CUE-WORD CONJ. *but*	0.10
CUE-WORD INSERT. *however*, CUE-WORD INSERT. *on the other hand*	0.09
PERIOD	0.01

Figure 14: Probability distribution of aggregation operators. The final operation is randomly chosen from the selected set with a uniform distribution.

feature sets tested for training the SPR. For example, the label at the root of the tree in Figure 15 (**CW-SINCE-NS-justify**) specifies that the CW-CONJUNCTION operation was used, with the *since* cue word, with the nucleus first (NS), to realize the *justify* relation. Similarly, the bottom left-most interior node (**WITH-NS-infer**) indicates that the WITH-REDUCTION operation was used, with the nucleus before the satellite (NS), to realize the *infer* relation.

Figure 17 shows a d-tree for the content plan in Figure 4. This d-tree shows that the SPG treats the PERIOD operation as part of the lexico-structural representation for the d-tree. The d-tree is split into multiple d-trees at these nodes before being sent to RealPro for surface realization.

Note that a tp-tree can have very different realizations, depending on the operations of the SPG. For example, the tp-tree in Figure 9 yields both Alt 6 and Alt 2 in Figure 1. Alt

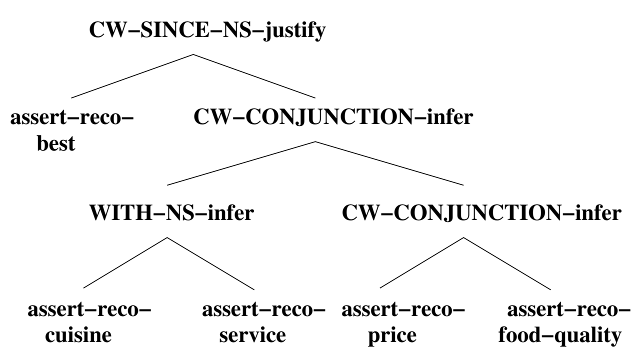

Figure 15: Sentence Plan Tree (SP-tree) for Alternative 6 of Figure 1.

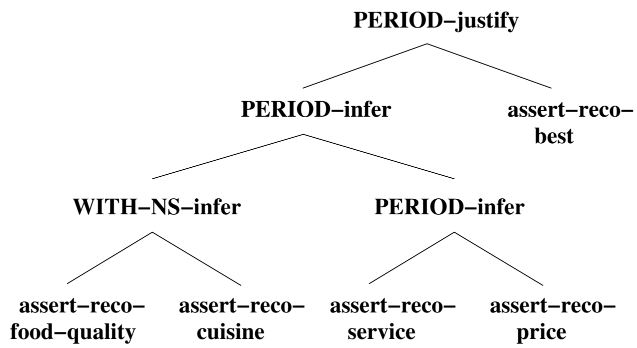

Figure 16: Sentence Plan Tree (SP-tree) for Alternative 8 of Figure 1.

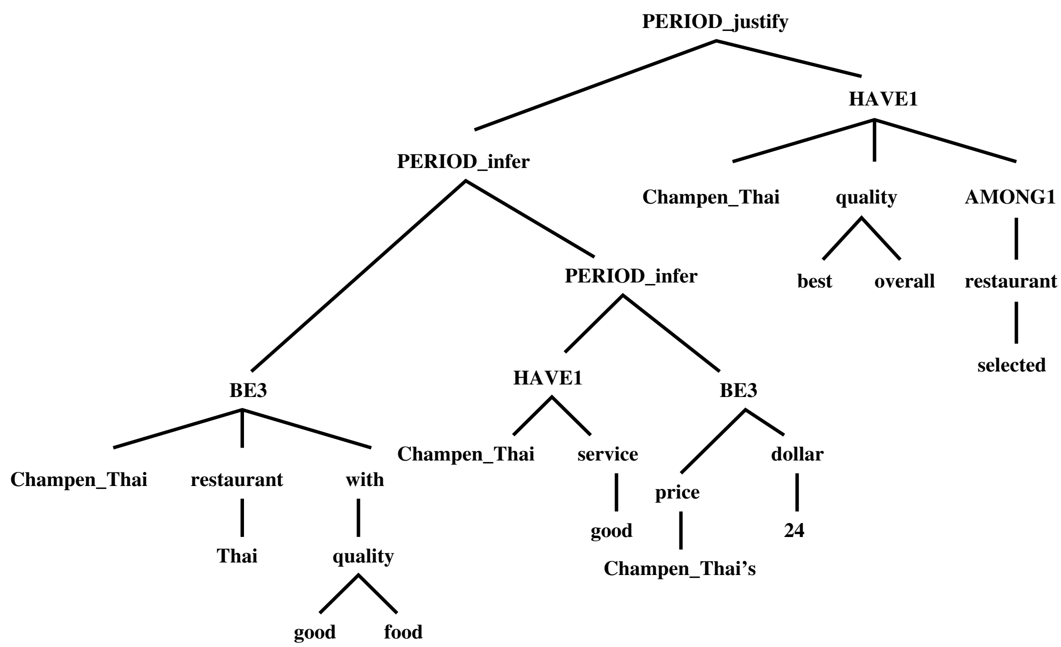

Figure 17: Dependency tree for alternative 8 in Figure 1.

2 is highly rated, with an average human rating of 4. However, Alt 6 is a poor realization of this plan, with an average human rating of 2.5.

To summarize, SPaRKy's SPG transforms an input content plan into a set of alternative pairs of sentence-plan trees and d-trees. First, assertions in the input content plan are grouped using principles from centering theory. Second, assertions are assigned to sentences and discourse cues inserted using clause combining operations. Third, decisions about the realization of referring expressions are made on the basis of recency. The rhetorical relations and clause-combining operations are domain-independent.

SPaRKy uses two types of domain-dependent knowledge: the probability distribution over clause-combining operations, and the d-trees that are input to the RealPro surface realizer. In order to use SPaRKy in a new domain, it might be necessary to:

- add new rhetorical relations if the content planner used additional rhetorical relations;
- modify the probability distribution over clause-combining operations, either by hand or by learning from a corpus;
- construct a new set of d-trees to capture the syntactic structure of sentences in the domain, unless we used a surface realizer that could take logical forms or semantic representations as input.

5. Feature Generation

To train or use the SPR, each potential realization generated by the SPG, along with its corresponding sp-tree and d-tree, is encoded as a set of real-valued features (binary features are modeled with values 0 and 1) from three feature types:

- **N-Gram features** – simple word n-gram features generated from the realization of SPG outputs;
- **Concept features** – concept n-gram features generated from named entities in the realization of SPG outputs;
- **Tree features** – these features represent structural configurations in the sp-trees and d-trees output by the SPG.

These features are automatically generated as described below.

5.1 N-Gram Features

N-gram features capture information about lexical selection and lexical ordering in the realizations output by SPaRKy. A two-step approach is used to generate these features. First, a domain-specific rule-based named-entity tagger (using MATCH's lexicons for restaurant, cuisine type and location names) replaces specific tokens with their types, e.g. *Babbo* with RESTNAME. Then, unigram, bigram and trigram features and their counts are automatically generated. The tokens *begin* and *end* indicate the beginning and end of a realization.

N-gram feature names are prefixed with N-GRAM. For example, NGRAM-CUISINENAME-RESTAURANT-WITH counts the occurrences of cuisine type followed by "restaurant" and "with" (as in the realization "Italian restaurant with"); NGRAM-BEGIN-RESTNAME-WHICH counts occurrences of realizations starting with a restaurant's name followed by "which". We also count words per presentation, and per sentence in a presentation.

5.2 Concept Features

Concept features capture information about the concepts selected for a presentation, and their linear order in the realization. A two-step approach is used to generate these features. First, a named-entity tagger marks the names of items in our restaurant database, e.g. *Uguale*. Then, unigram, bigram and trigram features and their counts are automatically generated from the sequences of concepts in the sentence plan for the realization. As with the n-gram features, the tokens *begin* and *end* indicate the beginning and end of a realization.

Concept feature names are prefixed with conc. For example, conc-decor-claim is set to 1 if the claim is expressed directly after information about decor, while the feature conc-begin-service characterizes utterances starting with information about service. In the concept n-gram features, we use '*' to separate individual features. We also count concepts per presentation, and per sentence in a presentation.

5.3 Tree Features

Tree features capture declaratively the way in which merge, infer and cue-word operations are applied to the tp-trees, and were inspired by the parsing features used by Collins (2000). They count the occurrences of certain structural linguistic configurations in the sp-trees and associated d-trees that the SPG generated. Tree feature names are prefixed with r for "rule" (sp-tree) or s for "sentence" (d-tree).

Several feature templates are used to generate tree features. *Local feature templates* record structural configurations local to a particular node (its ancestors, daughters etc.); *global feature templates*, used only for sp-tree features, record properties of the entire sp-tree.

There are four types of local feature template: traversal features, sister features, ancestor features and leaf features. Traversal, sister and ancestor features are generated for all nodes in sp-trees and d-trees; leaf features are generated for sp-trees only. The value of each feature is the count of the described configuration in the tree. We discard features that occur fewer than 10 times to avoid those specific to particular content plans.

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. Feature names are the concatenation of the prefix trav-, with the names of the nodes (starting with the current node) on the traversal path. '*' is used to separate node names. An example is r-trav-with-ns-infer*assert-reco-food-quality*assert-reco-cuisine (with value 1) of the bottom-left subtree in Figure 16.

Sister features record all consecutive sister nodes. Names are the concatenation of the prefix sis-, with the names of the sister nodes. An example is r-sis-assert-reco-best*cw-conjunction-infer (with value 1) of the tree in Figure 15.

For each node in the tree, **ancestor features** record all the initial subpaths of the path from that node to the root. Feature names are the concatenation of the prefix anc- with the names of the nodes (starting with the current node). An example is r-anc-assert-reco-cuisine*with-ns-infer*cw-conjunction-infer (with value 1) of the tree in Figure 15.

Leaf features record all initial substrings of the frontier of the sp-tree. Names are the concatenation of the prefix leaf-, with the names of the frontier nodes (starting with the current node). For example, the sp-tree of Figure 15 has value 1 for leaf-assert-reco-best and also for leaf-assert-reco-best*leaf-assert-reco-cuisine, and the sp-tree of Figure 16 has value 1 for leaf-assert-reco-food-quality*assert-reco-cuisine.

Global features apply only to the sp-tree. They record, for each sp-tree and for each operation labeling a non-frontier node, (1) the minimal number of leaves dominated by a node labeled with that rule in that tree (MIN); (2) the maximal number of leaves dominated by a node labeled with that rule (MAX); and (3) the average number of leaves dominated by a node labeled with that rule (AVG). For example, the sp-tree in Figure 15 has value

4 for cw-conjunction-infer-max, value 2 for cw-conjunction-infer-min and value 3 for cw-conjunction-infer-avg.

6. Training the Sentence Plan Ranker

The SPR ranks alternative information presentations using a model learned from user ratings of a set of training data. The training procedure is as follows:

- For each content plan in the training data, the SPG generates a set of alternative sentence plans using a random selection of sentence planning operators (Section 4);
- Features are automatically generated from the surface realizations and sentence plans so that each alternative sentence plan is represented in terms of a number of real-valued features (Section 5);
- Feedback as to the perceived quality of the realization of each alternative sentence plan is collected from one or more users;
- The RankBoost boosting method (Freund, Iyer, Schapire, & Singer, 1998) learns a function from the featural representation of each realization to its feedback, that attempts to duplicate the rankings in the training examples.

We use RankBoost for three reasons. First, it produces a ranking over the input alternatives rather than a selection of one best alternative. Second, it can handle many sparse features. Third, the function that it learns is a rule-based model showing the effect of each feature on the ranking of the competing examples. These models can be inspected and compared. This allows us to qualitatively analyze the models (Section 8) in order to understand the preferences of individuals, and the differences between SPRs for individuals vs. groups.

This section describes the training of the SPR in detail. The SPUR content planner produces content plans for three dialogue strategies:

- recommend: recommend an entity from a set of entities
- compare-2: compare two entities
- compare-3: compare three or more entities

For each dialogue strategy, we start with a set of 30 representative content plans from SPUR. The SPG was parameterized to produced up to 20 distinct (sp-tree, d-tree) pairs for each content plan. Each of these was realized by RealPro. Separately, we also obtained output for each content plan from our template-based generator (Section 3.2).

Both the SPaRKy realizations and the template-based realizations were randomly ordered and placed on a series of Web pages. These 1830 realizations were then rated on a scale from 1 to 5 by the first two authors of this paper, neither of whom had implemented the template-based realizer or the SPG. The raters worked on this rating task during sessions of one hour at a time for several hours a day, over a period of a week. They were instructed to look at all 21 realizations for a particular content plan before rating any of them, to try to use the whole rating scale, and to indicate their spontaneous rating without repeatedly re-labelling the alternative realizations. They did not discuss their ratings or the basis for their ratings at any time. Given the cognitive load and long duration of this rating task, it was impossible for the raters to keep track of which realizations came from SPaRKy and which from the template-based generator, and likely to be impossible to do more than generate a "gestalt" evaluation of each alternative.

Each (sp-tree, d-tree, realization) triple is an example input for RankBoost; the ratings are used as feedback. The experiments below examine two uses of the ratings. First, we train and test an SPR with the average of the ratings of the two users, i.e. we consider the two users as representing a single user group. Second, we train and test individualized SPRs, one for each user.

The SPR is trained using the RankBoost algorithm (Freund, Iyer, Schapire, & Singer, 1998), which we describe briefly here. First, the training corpus is converted into a set $\mathcal{T}$ of *ordered pairs* of examples x, y:

$$\begin{aligned} \mathcal{T} = \{(x,y)|\ & x, y \text{ are alternatives for the same plan,} \\ & x \text{ is preferred to } y \text{ by user ratings}\} \end{aligned}$$

Each alternative realization x is represented by a set of m indicator functions $h_s(x)$ for $1 \leq s \leq m$. The indicator functions are calculated by thresholding the feature values (counts) described in Section 5. For example, one indicator function is:

$$h_{100}(x) = \begin{cases} 1 & \text{if } \textsc{leaf-assert-reco-best}(x) \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

So $h_{100}(x) = 1$ if the leftmost leaf is the assertion of the claim as in Figure 15. A single parameter α_s is associated with each indicator function, and the "ranking score" for an example x is calculated as

$$F(x) = \sum_s \alpha_s h_s(x)$$

This score is used to rank competing sp-trees of the same content plan with the goal of duplicating the ranking found in the training data. Training is the process of setting the parameters α_s to minimize the following loss function:

$$RankLoss = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} eval(F(x) \leq F(y))$$

The *eval* function returns 1 if the ranking scores of the (x, y) pair are misordered (so that x is ranked higher than y even though in the training data y is ranked higher than x), and 0 otherwise. In other words, the RankLoss is the percentage of misordered pairs. As this loss function is minimized, the ranking errors (cases where ranking scores disagree with human judgments) are reduced. Initially all parameter values are set to zero. The optimization method then greedily picks a single parameter at a time – the parameter which will make the most impact on the loss function – and updates the parameter value to minimize the loss.

In the experiments described below, we use two evaluation metrics:

- **RankLoss**: The value of the training method's loss function;
- **TopRank**: The difference between the human rating of the top realization for each content plan and the human rating of the realization that the SPR predicts to be the top ranked.

7. Quantitative Results

In this section, we describe three experiments with SPaRKy:

1. **Feature sets for trainable sentence planning:** We examine which features (n-gram, concept, tree, all) lead to the best performance for the sentence planning task, and find that n-gram features sometimes perform as well as all the features.
2. **Comparison with template-based generation:** We show that the performance of a trainable sentence planner using the best performing feature set is more consistent than that of a template-based generator, although overall a template-based generator still performs better.
3. **Individualized sentence planners:** We show that people have quite specific individual preferences regarding the three tasks of sentence planning: information ordering, sentence aggregation, and use of discourse cues; and furthermore, that a trainable sentence planner can model these individual preferences. Moreover we show that in some cases the individualized sentence planners are better than, or statistically indistinguishable from, the template-based generator.

We report results below 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 produce text that is very different both in terms of length and structure.

7.1 Feature Sets for Trainable Sentence Planning

Using a cross-validation methodology, we repeatedly train the SPR on a random 90% of the corpus, and test on the remaining 10%. Here, we use the averaged feedback from user A and user B as feedback. Figure 18 repeats the examples in Figure 1, here showing both the user rankings and the rankings for a ranking function that was learned by the trained SPRs for both users A and B and for the AVG user.

Table 1 shows RankLoss for each feature set (Section 5). Paired t-tests comparing the ranking loss for different feature sets show surprisingly few performance differences among the features. Using all the features (**All**) always produces the best results, but the differences are not always significant.

The n-gram features give results comparable to all the features for both COMPARE-2 and RECOMMEND. An analysis of the learned models suggests that one reason that n-gram features perform well is because there are individual lexical items that are uniquely associated with many of the combination operators, such as the lexical item *with* for the WITH-NS operator. This means that the detailed representations of the content and structure of an information presentation as represented by the tree features are equivalent to n-gram features in this application domain.

Alt	Realization	A	B	SPR_A	SPR_B	SPR_{AVG}
6	Chanpen Thai has the best overall quality among the selected restaurants since it is a Thai restaurant, with good service, its price is 24 dollars, and it has good food quality.	1	4	0.16	0.65	0.58
7	Chanpen Thai has the best overall quality among the selected restaurants because it has good service, it has good food quality, it is a Thai restaurant, and its price is 24 dollars.	2	5	0.38	0.54	0.42
4	Chanpen Thai has the best overall quality among the selected restaurants. It has good food quality, with good service, it is a Thai restaurant, and its price is 24 dollars.	2	4	0.53	0.62	0.53
9	Chanpen Thai is a Thai restaurant, with good food quality, its price is 24 dollars, and it has good service. It has the best overall quality among the selected restaurants.	2	4	0.47	0.53	0.63
5	Chanpen Thai has the best overall quality among the selected restaurants. It has good service. It has good food quality. Its price is 24 dollars, and it is a Thai restaurant.	3	2	0.59	0.32	0.46
3	Chanpen Thai has the best overall quality among the selected restaurants. Its price is 24 dollars. It is a Thai restaurant, with good service. It has good food quality.	3	3	0.64	0.40	0.62
10	Chanpen Thai has the best overall quality among the selected restaurants. It has good food quality. Its price is 24 dollars. It is a Thai restaurant, with good service.	3	3	0.67	0.46	0.58
2	Chanpen Thai has the best overall quality among the selected restaurants. Its price is 24 dollars, and it is a Thai restaurant. It has good food quality and good service.	4	4	0.75	0.50	0.74
1	Chanpen Thai has the best overall quality among the selected restaurants. This Thai restaurant has good food quality. Its price is 24 dollars, and it has good service.	4	3	0.64	0.52	0.45
8	Chanpen Thai is a Thai restaurant, with good food quality. It has good service. Its price is 24 dollars. It has the best overall quality among the selected restaurants.	4	2	0.81	0.29	0.73

Figure 18: Some alternative realizations for the content plan in Figure 4, with feedback from users A and B (1=worst and 5=best) and rankings from the trained SPRs for users A and B and mean(A,B) ($[0, 1]$).

The concept features always perform worse than all the features, indicating that the linear ordering of concepts only accounts for some of the variation in rating feedback. For the two types of comparison, performance using the concept features approaches that of the other feature sets. However, for recommendations, performance using the concept features is much worse than that using n-gram features or all the features. In the qualitative analysis presented in Section 8, we discuss some aspects of the models for recommendations that might account for this large difference in performance.

Table 2 shows results with all the features using the TopRank evaluation metric, calculated for two-fold cross-validation, to be comparable with previous work (Walker, Rambow, & Rogati, 2002; Stent, Prasad, & Walker, 2004).[6] We evaluated SPaRKy on the test sets by comparing three data points for each content plan: Human (the score of the best sentence plan that SPaRKy's SPG can produce as selected by the human users); SPaRKy (the score of the SPR's top-ranked selected sentence); and Random (the score of a sentence plan

6. The TopRank metric is sensitive to the distribution of ranking feedback and SPR scores in the test set, which means that it is sensitive to the number of cross-validation folds.

Feature set/Strategy	COMPARE-2	COMPARE-3	RECOMMEND
Random Baseline	0.50	0.50	0.50
Concept	0.16 ($p < .000$)	0.16 ($p < .021$)	0.32 ($p < .000$)
N-Gram	**0.14** ($p < .161$)	0.15 ($p < .035$)	**0.21** ($p < .197$)
Tree	0.14 ($p < .087$)	0.16 ($p < .007$)	0.22 ($p < .001$)
All	**0.13**	**0.14**	**0.20**

Table 1: AVG model's ranking error with different feature sets, for all strategies. Results are averaged over 10-fold cross-validation, testing over the mean feedback. p values in parentheses indicate the level of significance of the decrease in accuracy when compared to the model using all the features. Cases where different feature sets perform as well as all the features are marked in bold.

randomly selected from the alternative sentence plans). 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, $p < .001$). The difference between the SPaRKy scores and the Human scores indicates how much performance could be improved if the SPR were perfect at replicating the Human ratings.

User	Strategy	SPaRKy	Human	Random
AVG	RECOMMEND	3.6 (0.77)	3.9 (0.55)	2.8 (0.81)
AVG	COMPARE-2	4.0 (0.66)	4.4 (0.54)	2.8 (1.30)
AVG	COMPARE-3	3.6 (0.68)	4.0 (0.49)	2.7 (1.20)

Table 2: TopRank scores for RECOMMEND, COMPARE-2 and COMPARE-3 (N = 180), using all the features, for SPaRKy trained on AVG feedback, with standard deviations.

7.2 Comparison with Template Generation

User	Strategy	SPaRKy	Human	Template
AVG	RECOMMEND	3.6 (0.59)	4.4 (0.37)	4.2 (0.74)
AVG	COMPARE-2	3.9 (0.52)	4.6 (0.39)	3.6 (0.75)
AVG	COMPARE-3	3.4 (0.38)	4.6 (0.35)	4.1 (1.23)

Table 3: TopRank scores for MATCH's template-based generator, SPaRKy(AVG) and Human. N = 180, with standard deviations.

As described above, the raters also rated the single output of the template-based generator for MATCH for each content plan in the training data. Table 3 shows the mean TopRank scores for the template-based generator's output (Template), compared to the best plan the trained SPR selects (SPaRKy), and the best plan as selected by a human oracle (Human). In each fold, both SPaRKy and the Human oracle select the best of 10 sentence plans for each text plan, while the template-based generator produces a single output with a single human-rated score. A paired t-test comparing Human with Template shows that there are no significant differences between them for RECOMMEND or COMPARE-3, but that Human is significantly better for COMPARE-2 ($df = 29$, $t = 4.8$, $p < .001$). The users evidently did not like the COMPARE-2 template. A paired t-test comparing SPaRKy to Template shows that the template-based generator is significantly better for both RECOMMEND and COMPARE-3 ($df = 29$, $t = 2.1$, $p < .05$), while there is a trend for SPaRKy to be better for COMPARE-2 ($df = 29$, $t = 2.0$, $p = .055$).

Also, the standard deviation for Template strategies is wider than for Human or SPaRKy, indicating that while the template-based generator performs well overall, it performs poorly on some inputs. One reason for this might be that SPUR's decision-theoretic user model selects a wide range and number of content items for different users, and for conciseness settings (See Figures 5 and 7). This means that it is difficult to handcraft a template-based generator to handle all the different cases well.

The gap between the Human scores (produced by the SPG but selected by a human rather than by the SPR) and the Template scores shows that the SPG produces sentence plans as good as those of the template-based generator, but the accuracy of the SPR needs to be improved. Below, Section 7.3 shows that when the SPR is trained for individuals, SPaRKy's performance is indistinguishable from the template-based generator in most cases.

7.3 Comparing Individualized Models to Group Models

We discussed in Section 1 that the differences in the rating feedback from users A and B for competing realizations (See Figure 1) suggest that each user has different perceptions as to the quality of the potential realizations. To quantify the utility and the feasibility of training individualized SPRs, we first examine the feasibility of training models for individual users.

The results in Table 1 are based on a corpus of 600 examples, rated by each user, which may involve too much effort for most users. We would like to know whether a high-performing individualized SPR can be trained from less labelled data. Figure 19 plots ranking error rates as a function of the amount of training data. This data suggests that error rates around 0.20 could be acquired with a much smaller training set, i.e. with a training set of around 120 examples, which is certainly more feasible.

RECOMMEND	A's model	B's model	AVG model
A's test data	0.17	0.52	0.29
B's test data	0.52	0.17	0.27
AVG's test data	0.31	0.31	0.20

Table 4: Ranking error for various configurations with the RECOMMEND strategy.

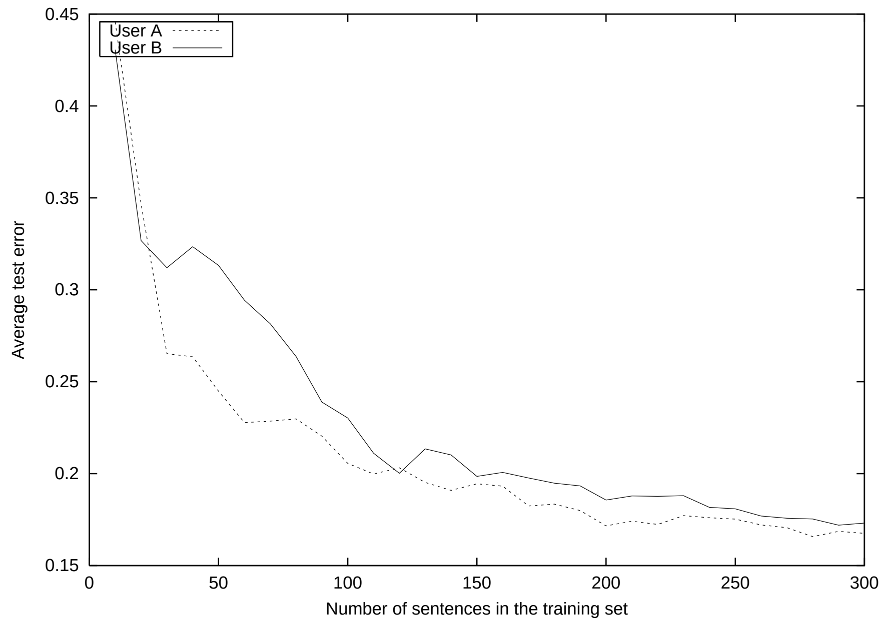

Figure 19: Variation of the testing error for both users as a function of the number of training utterances.

COMPARE-2	A's model	B's model	AVG model
A's test data	0.16	0.26	0.20
B's test data	0.23	0.11	0.13
AVG's test data	0.17	0.16	0.13

Table 5: Ranking error for various configurations with the COMPARE-2 strategy.

COMPARE-3	A's model	B's model	AVG model
A's test data	0.13	0.30	0.18
B's test data	0.26	0.14	0.18
AVG's test data	0.17	0.20	0.14

Table 6: Ranking error for various configurations with the COMPARE-3 strategy.

We then examine if trained individualized SPRs are accurate. The results in Tables 4, 5 and 6 show RankLoss for several training and testing configurations for each strategy (using 10-fold cross-validation). We compare the two individualized models with models trained on A and B's mean feedback (AVG). For each model, we test on its own test data, and on test data for the other models. This shows how well a model might 'fit' if customizing an SPR to a new domain or user group. For example, if we train a model for recommendations using feedback from a group of users, and then deploy this system to an individual user, we might expect model fit differences similar to those in Table 4.

Of course, there may be strongly conflicting preferences in any group of users. For example, consider the differences in the ratings for users A and B and the average ratings in Figure 1. Alt-1 and Alt-7 are equivalent using the average feedback, but user A dislikes Alt-7 and likes Alt-1 and vice versa for user B. Column 3 of Table 4 shows that the average model, when used in an SPR for user A or user B has a much higher ranking error (.29 and .27 respectively) than that of an SPR customized to user A (.17 error) or customized to user B (.17 error).

An examination of Tables 4, 5 and 6 shows that in general, there are striking differences between models trained and tested on one individual's feedback (RankLoss ranges from 0.11 to 0.17) and cross-tested models (RankLoss ranges from 0.13 to 0.52). Also, the average (AVG) models always perform more poorly for both users A and B than individually-tailored models. As a baseline for comparison, a model ranking sentence alternatives randomly produces an error rate of 0.5 on average; Table 4 shows that models trained on one user's data and tested on the other's can perform as badly as the random model baseline. This suggests that the differences in the users' ratings are not random noise.

In some cases, the average model also performs significantly worse than the individual models even when tested on feedback from the "average" user (the diagonal in Tables 4, 5 and 6). This suggests that in some cases it is harder to get a good model for the average user case, possibly because the feedback is more inconsistent. For recommendations, the performance of each individual model is significantly better than the average model ($df = 9$, $t = 2.6$, $p < .02$). For COMPARE-2 the average model is better than user A's ($df = 9$, $t = 2.3$, $p < .05$), but user B's model is better than the average model ($df = 9$, $t = 3.1$, $p < .01$).

User	Strategy	SPaRKy	Human	Template
A	RECOMMEND	3.5 (0.87)	3.9 (0.61)	3.9 (1.05)
A	COMPARE-2	3.8 (0.98)	4.3 (0.73)	4.2 (0.64)
A	COMPARE-3	3.1 (1.02)	3.6 (0.80)	3.9 (1.19)
B	RECOMMEND	4.4 (0.70)	4.7 (0.46)	4.5 (0.76)
B	COMPARE-2	4.4 (0.69)	4.7 (0.53)	3.1 (1.21)
B	COMPARE-3	4.4 (0.62)	4.8 (0.40)	4.2 (1.34)

Table 7: TopRank scores for the Individualized SPaRKy as compared with MATCH's template-based generator as rated separately by Users A and B, and individual User A and User B Human Oracles. Standard Deviations are in parentheses. N = 180.

We can also compare the template-based generator to the individualized SPaRKy generators using the TopRank metric (See Table 7). All comparisons are done with paired t-tests using the Bonferroni adjustment for multiple comparisons.

For RECOMMEND, there are no significant differences between SPaRKy and Template for User A ($df = 59$, $t = 2.3$, $p = .07$), or for User B ($df = 59$, $t = 1.6$, $p = .3$). There are also no significant differences for either user between Template and Human ($df = 59$, $t < 1.5$, $p > 0.4$).

For compare-2, there are large differences between Users A and B. User A appears to like the template for compare-2 (average rating is 4.2) while User B does not (average rating is 3.1). For User A, there are no significant differences between SPaRKy and Template ($df = 59$, $t = 2.3$, $p = .07$), and between Template and Human ($df = 59$, $t = 0.1$, $p = .09$), but User B strongly prefers SPaRKy to Template ($df = 59$, $t = 7.7$, $p < .001$).

For compare-3, there are also large differences between Users A and B. User A likes the template for compare-3 (average rating 3.9), and strongly prefers it to the individualized SPaRKy (average rating 3.1) ($df = 59$, $t = 3.4$, $p < .004$). User B also likes the template (average rating 4.2), but there are no significant differences with SPaRKy (average rating 4.4) ($df = 59$, $t = 1.0$, $p = .95$).

For both users, and for every strategy, even with individually trained SPRs, there is still a significant gap between SPaRKy and Human scores, indicating that the performance of the SPR could be improved ($df = 59$, $t = 3.0$, $p < .006$).

These results demonstrate that *trainable sentence planning can produce output comparable to or better than that of a template-based generator*, with less programming effort and more flexibility.

8. Qualitative Analysis

An important aspect of RankBoost is that the learned models are expressed as rules: a qualitative examination of the learned models may highlight individual differences in linguistic preferences, and help us understand why SPaRKy's SPG can produce sentence plans that are better than those produced by the template-based generator, and why the individually trained SPRs usually select sentence plans that are as good as the templates. To qualitatively compare the learned ranking models for the individualized SPRs, we assess both which linguistic aspects of an utterance (which features) are important to an individual, and how important they are. We evaluate whether an individual is oriented towards a particular feature by examining which features' indicator functions $h_s(x)$ have non-zero values. We evaluate how important a feature is to an individual by examining the magnitude of the parameters α_s.

There are two potential problems with this approach, The first problem is that the feature templates produce thousands of features, some of which are redundant, so that differences in each model's indicator functions can be spurious. Therefore, to allow more meaningful qualitative comparisons between models, one of a pair of perfectly correlated features is eliminated.

The second problem arises from RankBoost's greedy algorithm. The selection of which parameter α_s to set on any round of boosting is highly dependent on the training set, so that the models derived from a single episode of training are highly variable. To compare indicator functions independently of the training set, we adopt a bootstrapping method to identify a feature set for each user that is independent of a particular training episode. By repeatedly randomly selecting 10 alternatives for training and 10 for testing for each content plan, we created 50 different training sets for each user. We then average the α values of the features selected by RankBoost over these 50 training runs, and conduct experiments using

Model	Strategy	Feature Type				
		Tree	N-Gram	Concept	Leaf	Global
AVG	RECOMMEND	45	36	9	7	3
	COMPARE-2	37	46	12	1	4
	COMPARE-3	63	29	4	1	3
A	RECOMMEND	50	29	14	4	3
	COMPARE-2	35	51	10	3	1
	COMPARE-3	47	37	11	1	4
B	RECOMMEND	47	34	9	6	4
	COMPARE-2	45	36	13	1	5
	COMPARE-3	47	34	9	6	4

Table 8: Features in the top 100 with the highest average α for each user model.

only the 100 features for each user with the highest average α magnitude. In Section 8.1 we discuss differences in the types of feature that are selected by the bootsrapping algorithm just outlined. Section 8.2 discusses differences in models produced using the tree features for user A and user B, while section 8.3 discusses differences between the average model and the individual models.

8.1 Types of Bootstrapped Features

The bootstrapping process selects a total of 100 features for each strategy and for each type of feedback (individual or averaged). We found differences in the features along both dimensions.

Table 8 shows the number of features of each type that were in the top 100 (averaged over 50 training runs). Only 9 features are shared by the three strategies for the AVG model; these shared features are usually n-gram features. For User A, 6 features are shared by the three strategies (mostly n-gram features). For User B, there are no features shared by the three strategies.

We also found that some features capture specific interactions between domain-specific content items and syntactic structure, which are difficult to model in a rule-based or template-based generator. An example is Rule (1) in Figure 20 which significantly lowers the ranking of any sentence plan in which neighborhood information (ASSERT-RECO-NBHD) is combined with subsequent content items via the WITH-NS operation. Among the bootstrapped features for the average user, 16 features for COMPARE-2 count interactions between domain-specific content and syntactic structure. For COMPARE-3, 22 features count such interactions, and the bootstrapped features for RECOMMEND include 39 such features. We examine some of the models derived from these features in detail below.

8.2 Differences in Individual Models

To further analyze individual linguistic preferences for information presentation strategies, we now qualitatively compare the two models for Users A and B. We believe that this qualitative analysis provides additional evidence that the differences in the users' ranking preferences are not random noise. We identify differences among the features selected by

N	Condition	α
1	r-anc-assert-reco-nbhd*with-ns-infer ≥ 1	-1.26
2	cw-conjunction-infer-avg-leaves-under ≥ 3.1	-0.58
3	r-anc-assert-reco*with-ns-infer*cw-conjunction-infer ≥ 1	-0.33
4	leaf-assert-reco-best*assert-reco-price ≥ 1	-0.29
5	cw-conjunction-infer-avg-leaves-under ≥ 2.8	-0.27
6	r-trav-with-ns-infer*assert*assert ≥ 1	-0.22
7	r-anc-cw-conjunction-infer*cw-conjunction-infer ≥ 1	-0.17
8	with-ns-infer-min-leaves-under ≥ 1	-0.13
9	r-anc-assert-reco*with-ns-infer ≥ 1	-0.11
10	cw-conjunction-infer-max-leaves-under ≥ 3.5	-0.07
11	r-trav-with-ns-infer*assert-reco*assert-reco ≥ 1	-0.07
12	r-anc-assert*with-ns-infer ≥ 1	-0.03
13	r-anc-with-ns-infer*relative-clause-infer ≥ 1	-0.01
14	r-anc-assert*with-ns-infer*relative-clause-infer ≥ 1	-0.01
15	cw-conjunction-infer-avg-leaves-under ≥ 4.1	-0.01
16	r-anc-assert-reco-cuisine*with-ns-infer*period-infer ≥ 1	0.10
17	cw-conjunction-infer-avg-leaves-under ≥ 2.2	0.15
18	r-anc-assert-reco-food-quality*merge-infer ≥ 1	0.18
19	r-anc-assert-reco*merge-infer ≥ 2.5	0.20
20	r-anc-assert-reco-decor*merge-infer ≥ 1	0.22
21	r-anc-assert*merge-infer ≥ 2.5	0.25
22	r-trav-merge-infer ≥ 1.5	0.27
23	r-trav-with-ns-infer*assert-reco-service*assert-reco-food-quality $\geq$ 1	0.40
24	leaf-assert-reco-food-quality*assert-reco-cuisine ≥ 1	0.46
25	cw-conjunction-infer-avg-leaves-under ≥ 3.8	0.46
26	leaf-assert-reco-food-quality ≥ 1	0.60
27	s-trav-have1*propernoun-restaurant*II-quality*attr-among1 ≥ 1	0.68
28	s-anc-attr-with*have1 ≥ 1	0.71

Figure 20: A subset of rules and corresponding α values of User A's model, ordered by α.

RankBoost, and their α values, using models derived using bootstrapping over the tree features only, since they are easier to interpret qualitatively. Of course many different models are possible. User A's model consists of 109 rules; a subset are in Figure 20. User B's model consists of 90 rules, a subset of which are shown in Figure 21. We first consider how the individual models account for the rating differences for Alt-6 and Alt-8 from Figure 1 (repeated in Figure 18 with ratings from the trained SPRs), and then discuss other differences.

Comparing Alt-6 and Alt-8: Alt-6 is highly ranked by User B but not by User A. Alt-6 instantiates Rule 21 of Figure 21, expressing User B's preferences about linear order of the content. (Alt-6's sp-tree is in Figure 15.) Rule 21 increases the rating of examples in which the claim, i.e. assert-reco-best (*Chanpen Thai has the best overall quality*), is realized first. Thus, unlike user A, user B prefers the claim at the beginning of the utterance (the ordering of the claim is left unspecified by argumentation theory (Carenini & Moore, 2000)). Rule 22 increases the rating of examples in which the initial claim is immediately

N	Condition	α
1	R-SIS-ASSERT-RECO-RELATIVE-CLAUSE-INFER ≥ 1	-1.01
2	R-SIS-PERIOD-INFER-ASSERT-RECO ≥ 1	-0.71
3	R-ANC-ASSERT-RECO-NBHD*WITH-NS-INFER ≥ 1	-0.50
4	R-ANC-ASSERT-RECO*PERIOD-INFER*PERIOD-INFER ≥ 1.5	-0.49
5	R-ANC-ASSERT-RECO-FOOD-QUALITY*WITH-NS-INFER*RELATIVE-CLAUSE-INFER ≥ 1	-0.41
6	R-ANC-ASSERT-RECO-CUISINE*WITH-NS-INFER*RELATIVE-CLAUSE-INFER ≥ 1	-0.39
7	R-ANC-ASSERT-RECO*PERIOD-INFER ≥ 1	-0.35
8	LEAF-ASSERT-RECO-PRICE ≥ 1	-0.32
9	R-ANC-ASSERT*PERIOD-INFER*PERIOD-INFER ≥ 1.5	-0.26
10	LEAF-ASSERT-RECO-DECOR ≥ 1	-0.14
11	R-ANC-ASSERT*RELATIVE-CLAUSE-INFER*PERIOD-INFER ≥ 1.5	-0.07
12	R-TRAV-RELATIVE-CLAUSE-INFER*ASSERT-RECO*WITH-NS-INFER ≥ 1	-0.05
13	CW-CONJUNCTION-INFER-AVG-LEAVES-UNDER ≥ 3.1	-0.03
14	CW-CONJUNCTION-INFER-AVG-LEAVES-UNDER ≥ 3.3	-0.03
15	CW-CONJUNCTION-INFER-AVG-LEAVES-UNDER ≥ 2.2	-0.01
16	R-ANC-ASSERT*RELATIVE-CLAUSE-INFER*PERIOD-INFER ≥ 1	0.03
17	LEAF-ASSERT-RECO-SERVICE ≥ 1	0.07
18	S-TRAV-ATTR-WITH ≥ 1	0.18
19	R-ANC-ASSERT-RECO-CUISINE*WITH-NS-INFER*CW-CONJUNCTION-INFER ≥ 1	0.27
20	CW-CONJUNCTION-INFER-AVG-LEAVES-UNDER ≥ 3.6	0.36
21	LEAF-ASSERT-RECO-BEST ≥ 1	0.47
22	LEAF-ASSERT-RECO-BEST*ASSERT-RECO-CUISINE ≥ 1	0.50
23	CW-CONJUNCTION-INFER-AVG-LEAVES-UNDER ≥ 2.8	0.52
24	R-TRAV-WITH-NS-INFER*ASSERT-RECO-CUISINE*ASSERT-RECO-FOOD-QUALITY ≥ 1	0.76

Figure 21: A subset of rules and corresponding α values of User B's model, ordered by α.

followed by the type of cuisine (ASSERT-RECO-CUISINE). These rules interact with Rule 19 in Figure 21, which specifies a preference for information following ASSERT-RECO-CUISINE to be combined via the WITH-NS operation, and then conjoined (CW-CONJUNCTION-INFER) with additional evidence. Alt-6 also instantiates Rule 23 in User B's model, with an α value of .52 associated with multiple uses of the CW-CONJUNCTION-INFER operation.

User A's low rating of Alt-6 arises from A's dislike of the WITH-NS operation (Rules 3, 8, 9, 11 and 12) and the CW-CONJUNCTION-INFER operation (Rules 3, 5, 7, 10 and 15) in Figure 20. (Contrast User B's Rule 23 with User A's Rules 5 and 17.) Alt-6 also fails to instantiate A's preference for food quality and cuisine information to occur first (Rules 24 and 26). Finally, user A also prefers the claim ASSERT-RECO-BEST to be realized in its own sentence (Rule 27).

By contrast, **Alt-8** is rated highly by User A but not by User B (see Figure 1). Even though Alt-8 instantiates the negatively evaluated WITH-NS operation (Rules 3, 8, 9 and 11 in Figure 20), there are no instances of CW-CONJUNCTION-INFER (Rules 3, 5, 7, 10 and 15). Moreover Alt-8 follows A's ordering preferences (Rules 24 and 26) which describe sp-trees with ASSERT-RECO-FOOD-QUALITY on the left frontier, and trees where it is followed by ASSERT-RECO-CUISINE. (See Alt-8's sp-tree in Figure 16.) Rule 27 also increases the rating of Alt-8 with its large positive α reflecting the expression of the claim in its own sentence.

On the other hand, Alt-8 is rated poorly by User B; it violates B's preferences for linear order (remember that Rules 21 and 22 specify that B prefers the claim first, followed by cuisine information). Also, B's model has rules that radically decrease the ranking of examples using the PERIOD-INFER operation (Rules 2, 4, 7 and 9).

Thus, Alt-6 and Alt-8 show that users A and B prefer different combination operators, and different ordering of content, e.g. B likes the claim first and A likes recommendations with food quality first followed by cuisine. As mentioned above, previous work on the generation of evaluative arguments states that the claim may appear first or last (Carenini & Moore, 2000). The relevant guideline for producing effective evaluative arguments states that "placing the main claim first helps users follow the line of reasoning, but delaying the claim until the end of the argument can also be effective if the user is likely to disagree with the claim." The template-based generator for MATCH always placed the claim first, but this analysis suggests that this may not be effective for user A.

Other similarities and differences: There are also individual differences in preferences for particular operations, and for specific content operation interactions. For example, User A's model demotes examples where the WITH-NS operation has been applied (Rules 3, 6 and 8 in Figure 20), while User B generally likes examples where WITH-NS has been used (Rule 18 in Figure 21). However, neither A nor B like WITH-NS when used to combine other content with neighborhood information. In User A's model the α value is -1.26, while in User B's model the value is -0.50 (see Rule 1 in Figure 20 and Rule 3 in Figure 21.) These rules capture a specific interaction in the sp-tree between domain-specific content and the WITH-NS-INFER combination operation. Utterances instantiating these rules place information in an adjunctival with-clause following the clause realizing the restaurant's neighborhood. There is no constraint on the type of information in the with-clause. In utterance (1) below, the with-clause realizes the restaurant's food quality, whereas in (2) it contains information about the restaurant's service.

(1) Mont Blanc has very good service, its price is 34 dollars, and it is located in Midtown West, with good food quality. It has the best overall quality among the selected restaurants.

(2) Mont Blanc is located in Midtown West, with very good service, its price is 34 dollars, and it has good food quality. It has the best overall quality among the selected restaurants.

Moreover, both users like WITH-NS when it combines cuisine and food-quality information as in example (3) (Rule 23 in Figure 20 and Rule 24 in Figure 21).

(3) Komodo has the best overall quality among the selected restaurants since it is a Japanese, Latin American restaurant, with very good food quality, it has very good service, and its price is 29 dollars.

But User B radically reduces the rating of the cuisine, food-quality combination when it is combined with further information using the RELATIVE-CLAUSE-INFER operation, as in example (5) (Rules 5 and 6 in Figure 21).

(4) Bond Street has very good decor. This Japanese, Sushi restaurant, with excellent food quality, has good service. It has the best overall quality among the selected restaurants.

Example (4) is an interesting contrast with example (3). Example (4) instantiates Rule 24 in Figure 21, but it also instantiates a number of negatively valued features. As discussed above, User B prefers examples where the claim is expressed first (Rule 21 in Figure 21), and User B's model explicitly reduces the rating of examples where information about decor is expressed first (Rule 10 in Figure 21).

In general, User A likes the MERGE-INFER operation (Rules 19, 21 and 22), especially when applied with ASSERT-RECO-FOOD-QUALITY (Rule 18), and ASSERT-RECO-DECOR (Rule 20). User A strongly prefers to hear about food quality first (Rule 26 in Figure 20), followed by cuisine information (Rule 24). In contrast, User B has rules that reduce the rating of examples with price or decor first (Rules 8 and 10 in Figure 21). User B also has no preferences for MERGE-INFER but likes the CW-CONJUNCTION operation (Rule 20 in Figure 21). Finally, User B dislikes the RELATIVE-CLAUSE-INFER operation in general (Rule 1), and its combination with the WITH-NS operation (Rule 12) or the PERIOD-INFER operation (Rule 11).

In addition to other evidence discussed above as to individual differences in language generation, we believe that the fact that these model differences are *interpretable* shows that the differences in user perception of the quality of system utterances are true individual differences, and not random noise.

8.3 Average Model Differences

Table 22 shows a subset of rules that have the largest α magnitudes for an example AVG model using the same 100 feature bootstrapping process described above. Section 8.2 presented results that the average model performs statistically worse for recommendations than either of the individual models. This may be due to the fact that the average model is essentially trying to learn from contradictory feedback from the two users. To see whether an examination of the models provides support for this hypothesis, we first examine how the learned model ranks Alt-6 And Alt-8 as shown in Figure 18 in the column SPR_{AVG}. The average feedback for Alt-6 is 2.5 while the average feedback for Alt-8 is 3, but the trained SPR ranks Alt-8 second highest and Alt-6 fifth out of 10.

The mid-value ranking of Alt-6 arises from a number of interacting rules, some of which are similar to User B's and some of which are similar to User A's. Alt-6 instantiates Rules 26 and 27 in Figure 22 which increase the ranking of sentence plans in which the claim, i.e. ASSERT-RECO-BEST is realized first, and sentence plans where the claim is immediately followed by information about the type of cuisine (ASSERT-RECO-CUISINE). These rules are identical to B's Rules 21 and 22 in Figure 21. Rule 18 additionally increases the ranking of sentence plans where cuisine information is followed by service information, which applies to Alt-6 to further increase its ranking. However Rule 3 lowers the ranking of Alt-6, since it combines more than 3 different assertions into a single DSyntS tree.

Alt-8 is highly ranked by SPR_{AVG}, largely as a result of several rules that increase its ranking. Rule 31 specifies an increase in ranking for sentence plans that have the claim in its own sentence, which is true of Alt-8 but not of Alt-6. This rule also appears as Rule 27

N	Condition	α_s
1	s-anc-attr-with*locate $\geq -\infty$	-0.87
2	s-trav-have1*i-restaurant*cuisine-type*ii-quality*attr-good*attr-food $\geq -\infty$	-0.81
3	s-trav-propernoun-restaurant ≥ 2.5	-0.81
4	r-anc-cw-conjunction-infer*cw-conjunction-infer*period-justify $\geq -\infty$	-0.77
5	r-sis-assert-reco-relative-clause-infer $\geq -\infty$	-0.74
6	r-anc-assert-reco-decor*with-ns-infer*period-infer*period-justify $\geq -\infty$	-0.62
7	r-anc-assert-reco-cuisine*with-ns-infer*relative-clause-infer $\geq -\infty$	-0.62
8	period-justify-avg-leaves-under ≥ 5.5	-0.60
9	cw-conjunction-infer-avg-leaves-under ≥ 3.1	-0.54
10	r-anc-assert-reco-nbhd*with-ns-infer $\geq -\infty$	-0.45
11	r-sis-cw-conjunction-infer-relative-clause-infer $\geq -\infty$	-0.40
12	period-infer-avg-leaves-under ≥ 3.4	0.14
13	r-anc-assert-reco-food-quality*merge-infer $\geq -\infty$	0.15
14	s-trav-propernoun-restaurant ≥ 5.5	0.19
15	r-anc-assert-reco-decor*merge-infer $\geq -\infty$	0.19
16	s-anc-attr-with*i-restaurant*have1 $\geq -\infty$	0.22
17	r-anc-assert-reco-decor*with-ns-infer $\geq -\infty$	0.26
18	leaf-assert-reco-best*assert-reco-cuisine*assert-reco-service $\geq -\infty$	0.26
19	s-trav-propernoun-restaurant ≥ 3.5	0.29
20	r-anc-assert-reco-cuisine*with-ns-infer*period-infer*period-justify $\geq -\infty$	0.29
21	leaf-assert-reco-food-quality $\geq -\infty$	0.32
22	period-infer-avg-leaves-under ≥ 3.2	0.36
23	r-sis-merge-infer-assert-reco $\geq -\infty$	0.42
24	period-justify-avg-leaves-under ≥ 6.5	0.48
25	s-anc-attr-with*have1 $\geq -\infty$	0.49
26	leaf-assert-reco-best*assert-reco-cuisine $\geq -\infty$	0.50
27	leaf-assert-reco-best $\geq -\infty$	0.50
28	merge-infer-max-leaves-under $\geq -\infty$	0.51
29	leaf-assert-reco-food-quality*assert-reco-cuisine $\geq -\infty$	0.77
30	merge-infer-max-leaves-under ≥ 2.5	0.96
31	s-trav-have1*propernoun-restaurant*ii-quality*attr-among1 $\geq -\infty$	0.97

Figure 22: A subset of the rules with the largest α magnitudes that were learned for ranking recommendations given AVG feedback.

in A's model in Figure 20. Alt-8 also instantiates Rules 21 and 29 which which are identical to user A's ordering preferences (Rules 24 and 26 in Figure 20) These rules describe sp-trees with assert-reco-food-quality on the left frontier, and trees where it is followed by assert-reco-cuisine. (See Alt-8's sp-tree in Figure 16.) Rule 3 also applies to Alt-8, reducing its ranking due to the number of content items it realizes.

Other similarities and differences: There are many rules in the average model that are similar to either A or B's models or both, and the average model retains a number of preferences seen in the individual models. For example, Rules 1 and 10 both reduce the ranking of any sentence plan where neighborhood information is combined with subsequent information using the WITH-NS combination operator. Rule 1 expresses this in terms of the lexical items in the d-tree, whereas Rule 10 expresses it in terms of semantic features derived from the sp-tree. Examples 1 and 2 in Section 8.2 illustrate this interaction.

Some of the rules are more similar to User A. For example, Rules 4 and 9 (like A's Rules 2 and 5 in Figure 20) reduce the rating of sentence plans that use the operation CW-CONJ-INFER. In addition, Rules 22, 23, 24, and 28 expresses preferences for merging information, which are very similar to A's Rules 19, 21 and 22. Rule 15 expresses a preference for information about the atmosphere (ASSERT-RECO-DECOR) to be combined using the MERGE operation, as specified in A's Rule 20. Rule 20 in Figure 22 is also similar to A's Rule 16 with ASSERT-RECO-CUISINE combined with subsequent information with the WITH-NS operation.

Other rules are more similar to B's model. For example, Rule 5 reduces the ranking of sentence plans using the RELATIVE CLAUSE operation, which was also specified in User B's Rule 1, and Rules 16 and 25 indicate a general preference for use of the with-ns operation, which was a strong preference in User B's model (see B's Rule 18 in Figure 21).

Note that in some cases, the learned model tries to account for both A's and B's preferences, even when these contradict one another. For example, Rule 27 specifies a preference for the claim to come first, as in B's Rule 21, whereas Rule 26 is the same as A's 24, specifying a preference for food quality and cuisine information to be expressed first. Thus the model does suggest that a reduction in performance may arise from trying to account for the contradictory preferences of users A and B.

9. Conclusions

This article describes SPaRKy, a two-stage sentence planner that generates many alternative realizations of input content plans and then ranks them using a statistical model trained on human feedback. We demonstrate that the training technique developed for SPoT (Walker, Rambow, & Rogati, 2002), generalizes easily to new domains, and that it can be extended to handle the rhetorical structures required for more complex types of information presentation.

One of the most novel contributions of this paper is to show that trainable generation can be used to train sentence planners tailored to individual users' preferences. Previous work modeling individuals has mainly applied to content planning. While studies of human-human dialogue suggest that modeling other types of individual differences could be valuable for spoken language generation, in the past, linguistic variation among individuals was considered a problem for generation (McKeown, Kukich, & Shaw, 1994; Reiter, 2002; Reiter, Sripada, & Robertson, 2003). Here, we show that users have different perceptions of the quality of alternative realizations of a content plan, and that individualized models perform better than those trained for groups of users. Our qualitative analysis indicates that trainable sentence generation is sensitive to variations in domain application, presentation type, and individual human preferences about the arrangement of particular content types. These are the first results showing that individual preferences apply to sentence planning.

We also compared SPaRKy to the template-based generator described in Section 3.2: this generator is highly tuned to this domain and was previously shown to produce high quality outputs in a user evaluation (Stent, Prasad, & Walker, 2004). When SPaRKy is trained for a group of users, then template-based generation is better for recommend and compare-3, but in most cases the performance of the individualized SPRs are statistically indistinguishable from MATCH's template-based generator: the exceptions are that, for compare-2, User B prefers SPaRKy, while for compare-3 User A prefers the template-based generator. In all cases, the Human scores (outputs produced by the SPG but selected by a human) are as good or better than the template-based generator, even for complex information presentations such as extended comparisons.

These results show that there is a gap between the performance of the trained SPR and human performance. This suggests that it might be possible to improve the SPR with different feature sets or a different ranking algorithm. We leave a comparison with other ranking algorithms to future work. Here, we report results for many different feature sets (n-gram, concept and tree) and investigate their effect on performance. Table 1 shows that a combination of the three feature sets performs significantly better for recommend and compare-3 than the tree features from our earlier work (Walker, Rambow, & Rogati, 2002; Stent, Prasad, & Walker, 2004; Mairesse & Walker, 2005). Interestingly, in some cases, simple features like n-grams perform as well as features representing linguistic structure such as the tree features. This might be because particular lexical items, e.g. *with*, are often uniquely associated with a combination operator, e.g. the with-ns operator, which was shown to have impact on user perceptions of utterance quality (Section 8). More work is needed to determine whether these performance similarities are simply due to the fact that the variation of form generated by SPaRKy's SPG is limited. Other work has also examined tradeoffs between n-gram features and linguistically complex features in terms of tradeoffs between time and accuracy (Pantel, Ravichandran, & Hovy, 2004). Although SPaRKy is trained offline, the time to compute features and rank SPG outputs remains an issue when using SPaRKy in a real-time spoken dialogue system.

A potential limitation of our approach is the time and effort required to elicit user feedback for training the system, as described in Section 6. In Section 7.3 we showed that RankLoss error rates of around 0.20 could be acquired with a much smaller training set, i.e. with a training set of around 120 examples. However typical users would probably not want to provide ratings of 120 examples. Future work should explore alternative training regimes perhaps by utilizing ratings from several users. For example, we could identify examples that most distinguish our existing users, and just present these examples to new users. Also, instead of users rating information presentations before using MATCH, perhaps a method for users to rate information presentations while using MATCH could be developed, i.e. in the course of a dialogue with MATCH when a recommendation or comparison is presented to the user, the system could display on the screen a rating form for that presentation. Another approach would be to train from a different type of user feedback collected automatically by monitoring the user's behavior, e.g. measures of cognitive load such as reading time.

Another limitation is that SPaRKy's dictionary is handcrafted, i.e. the associations between simple assertions and their syntactic realizations (d-trees) are specified by hand, like all generators. Recent work has begun to address this limitation by investigating techniques for learning a generation dictionary automatically from different types of corpora, such as user reviews (Barzilay & Lee, 2002; Higashinaka, Walker, & Prasad, 2007; Snyder & Barzilay, 2007).

A final limitation is that we only use two individuals to provide a proof-of-concept argument for the value of user-tailored trainable sentence planning. We have argued throughout this paper that the individual differences we document are more general, are not particular to users A and B, and are not the result of random noise in user feedback. Nevertheless, we hope that future work will test these results against a larger population of individuals in order to provide further support for these arguments and in order to characterize the full range of individual differences in preferences for language variation in dialogue interaction.

Acknowledgments

This work was partially funded by a DARPA Communicator Contract MDA972-99-3-0003, by a Royal Society Wolfson Research Merit Award to M. Walker, and by a Vice Chancellor's studentship to F. Mairesse.

References

André, E., Rist, T., van Mulken, S., Klesen, M., & Baldes, S. (2000). *Embodied Conversational Agents*, chap. The automated design of believable dialogues for animated presentation teams, pp. 220–255. MIT Press.

Bangalore, S., & Rambow, O. (2000). Exploiting a probabilistic hierarchical model for generation. In *Proc. of the International Conference on Computational Linguistics*.

Barzilay, R., Elhadad, N., & McKeown, K. R. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, *17*, 35–55.

Barzilay, R., & Lee, L. (2002). Bootstrapping lexical choice via multiple-sequence alignment. In *Proc. of the Conference on Empirical Methods for Natural Language Processing*.

Belz, A. (2005). Corpus-driven generation of weather forecasts. In *Proc. 3rd Corpus Linguistics Conference*.

Bouayad-Agha, N., Scott, D., & Power, R. (2000). Integrating content and style in documents: a case study of patient information leaflets. *Information Design Journal*, *9*(2), 161–176.

Branigan, H., Pickering, M., & Cleland, A. (2000). Syntactic coordination in dialogue. *Cognition*, *75*, B13–B25.

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory And Cognition*, *22*(6), 1482–1493.

Brennan, S. E., Friedman, M. W., & Pollard, C. J. (1987). A centering approach to pronouns. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Brown, P., & Levinson, S. (1987). *Politeness: Some Universals in Language Usage*. Cambridge University Press.

Bulyko, I., & Ostendorf, M. (2001). Joint prosody prediction and unit selection for concatenative speech synthesis. In *Proc. of the International Conference on Acoustic Speech and Signal Processing*.

Carenini, G., & Moore, J. D. (2000). A strategy for generating evaluative arguments. In *Proc. of the International Natural Language Generation Conference*.

Carenini, G., & Moore, J. D. (2006). Generating and evaluating evaluative arguments. *Artificial Intelligence Journal*, *170*(11), 925–952.

Chai, J., Hong, P., Zhou, M., & Prasov, Z. (2004). Optimization in multimodal interpretation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Chu-Carroll, J., & Carberry, S. (1995). Response generation in collaborative negotiation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, *22*, 1–39.

Collins, M. (2000). Discriminative reranking for natural language parsing. In *Proc. of the International Conference on Machine Learning*.

Coulston, R., Oviatt, S., & Darves, C. (2002). Amplitude convergence in children's conversational speech with animated personas. In *Proc. of the International Spoken Language Processing Conference*.

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

Di Eugenio, B., Moore, J. D., & Paolucci, M. (1997). Learning features that predict cue usage. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

DiMarco, C., & Foster, M. E. (1997). The automated generation of Web documents that are tailored to the individual reader. In *Proc. of the AAAI Spring Symposium on Natural Language Processing on the World Wide Web*.

Duboue, P. A., & McKeown, K. R. (2003). Statistical acquisition of content selection rules for natural language generation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*.

Elhadad, N., Kan, M.-Y., Klavans, J., & McKeown, K. (2005). Customization in a unified framework for summarizing medical literature. *Journal of Artificial Intelligence in Medicine*, *33*(2), 179–198.

Ferguson, S. H., & Kewley-Port, D. (2002). Vowel intelligibility in clear and conversational speech for normal-hearing and hearing-impaired listeners. *Journal of the Acoustical Society of America*, *112*(1), 259–271.

Fleischman, M., & Hovy, E. (2002). Emotional variation in speech-based natural language generation. In *Proc. of the International Natural Language Generation Conference*.

Forbes, K., Miltsakaki, E., Prasad, R., Sarkar, A., Joshi, A., & Webber, B. (2003). D-LTAG system: Discourse parsing with a lexicalized tree adjoining grammar. *Journal of Logic, Language and Information*, *12*(3), 261–279.

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

Garrod, S., & Anderson, A. (1987). Saying what you mean in dialogue: A study in conceptual and semantic coordination. *Cognition*, *27*, 181–218.

Goffman, E. (1981). *Forms of Talk*. University of Pennsylvania Press, Philadelphia, Pennsylvania, USA.

Grosz, B. J., Joshi, A. K., & Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, *21*(2), 203–225.

Guo, H., & Stent, A. (2005). Trainable adapatable multimedia presentation generation. In *Proc. of the International Conference on Multimodal Interfaces*. Demo paper.

Gupta, S., Walker, M., & Romano, D. (2007). How rude are you?: Evaluating politeness and affect in interaction. In *Proc. of the Second International Conference on Affective Computing and Intelligent Interaction*.

Gupta, S., & Stent, A. (2005). Automatic evaluation of referring expression generation using corpora. In *Proc. of the Workshop on Using Corpora in Natural Language Generation*.

Hardt, D., & Rambow, O. (2001). Generation of VP ellipsis: A corpus-based approach. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Higashinaka, R., Walker, M., & Prasad, R. (2007). An unsupervised method for learning generation lexicons for spoken dialogue systems by mining user reviews. *ACM Transactions on Speech and Language Processing*, *4*(4).

Hovy, E. (1987). Some pragmatic decision criteria in generation. In Kempen, G. (Ed.), *Natural Language Generation*, pp. 3–17. Martinus Nijhoff.

Isard, A., Brockmann, C., & Oberlander, J. (2006). Individuality and alignment in generated dialogues. In *Proc.of the International Natural Language Generation Conference*.

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., & Maloor, P. (2002). MATCH: An architecture for multimodal dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Jokinen, K., & Kanto, K. (2004). User expertise modelling and adaptivity in a speech-based e-mail system. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Jordan, P., & Walker, M. (2005). Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, *24*, 157–194.

Joshi, A. K., Webber, B., & Weischedel, R. M. (1986). Some aspects of default reasoning in interactive discourse. Tech. rep. MS-CIS-86-27, University of Pennsylvania.

Joshi, A. K., Webber, B. L., & Weischedel, R. M. (1984). Preventing false inferences. In *Proc. of the International Conference on Computational Linguistics*.

Jungers, M. K., Palmer, C., & Speer, S. R. (2002). Time after time: The coordinating influence of tempo in music and speech. *Cognitive Processing*, *1–2*, 21–35.

Kittredge, R., Korelsky, T., & Rambow, O. (1991). On the need for domain communication knowledge. *Computational Intelligence*, *7*(4), 305–314.

Kothari, A. (2007). Accented pronouns and unusual antecedents: A corpus study. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*.

Langkilde, I., & Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proc. of the International Conference on Computational Linguistics and Meeting of the Association for Computational Linguistics*.

Langkilde-Geary, I. (2002). An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. of the International Natural Language Generation Conference*.

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Lavoie, B., & Rambow, O. (1997). A fast and portable realizer for text generation systems. In *Proc. of the Conference on Applied Natural Language Processing*.

Levelt, W. J. M., & Kelter, S. (1982). Surface form and memory in question answering. *Cognitive Psychology*, *14*, 78–106.

Lin, J. (2006). Using distributional similarity to identify individual verb choice. In *Proc. of the International Natural Language Generation Conference*.

Litman, D. (1996). Cue phrase classification using machine learning. *Journal of Artificial Intelligence Research*, *5*, 53–94.

Luchok, J. A., & McCroskey, J. C. (1978). The effect of quality of evidence on attitude change and source credibility. *The Southern Speech Communication Journal*, *43*, 371–383.

Madigan, D., Genkin, A., Lewis, D., Argamon, S., Fradkin, D., & Ye, L. (2005). Author identification on the large scale. In *Proc. of the Meeting of the Classification Society of North America*.

Mairesse, F., & Walker, M. (2007). PERSONAGE: Personality generation for dialogue. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Mairesse, F., & Walker, M. (2005). Learning to personalize spoken generation for dialogue systems. In *Proc. Interspeech*.

Mann, W., & Thompson, S. (1987). Rhetorical structure theory: Description and construction of text structures. In Kempen, G. (Ed.), *Natural Language Generation*, pp. 83–96. Martinus Nijhoff.

Marciniak, T., & Strube, M. (2004). Classification-based generation using TAG. In *Proc. of the International Natural Language Generation Conference*.

Marcu, D. (1996). Building up rhetorical structure trees. In *Proc. of the Conference on Artificial Intelligence and Conference on Innovative Applications of Artificial Intelligence*.

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

McCoy, K. F. (1989). Generating context-sensitive responses to object related misconceptions. *Artificial Intelligence*, *41*(2), 157–195.

McKeown, K., Kukich, K., & Shaw, J. (1994). Practical issues in automatic document generation. In *Proc. of the Conference on Applied Natural Language Processing*.

McKeown, K. R. (1985). Discourse strategies for generating natural language text. *Artificial Intelligence*, *27*(1), 1–42.

Mellish, C., O'Donnell, M., Oberlander, J., & Knott, A. (1998). An architecture for opportunistic text generation. In *Proc. of the Ninth International Workshop on Natural Language Generation*.

Melčuk, I. A. (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, New York.

Moore, J. D., & Paris, C. L. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, *19*(4), 651–694.

Moore, J. D., Foster, M. E., Lemon, O., & White, M. (2004). Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of the Seventeenth International Florida Artificial Intelligence Research Society Conference*.

Nakatsu, C., & White, M. (2006). Learning to say it well: Reranking realizations by predicted synthesis quality. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Nenkova, A., Passonneau, R. J., & McKeown, K. (2007). The pyramid method: incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, *4*(2).

Oberlander, J., & Brew, C. (2000). Stochastic text generation. *Philosophical Transactions of the Royal Society of London*, *Series A, 358*, 1373–1385.

Paiva, D. S., & Evans, R. (2004). A framework for stylistically controlled generation. In *Proc. of the International Natural Language Generation Conference*.

Pantel, P., Ravichandran, D., & Hovy, E. (2004). Towards terascale knowledge acquisition. In *Proc. of the International Conference on Computational Linguistics*.

Papenini, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Pennebaker, J. W., & King, L. A. (1999). Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, *77*, 1296–1312.

Piwek, P. (2003). A flexible pragmatics-driven language generator for animated agents. In *Proc. of the European Meeting of the Association for Computational Linguistics*.

Polifroni, J., & Walker, M. (2006). An analysis of automatic content selection algorithms for spoken dialogue system summaries. In *Proc. of the IEEE/ACL Conference on Spoken Language Technology*.

Porayska-Pomsta, K., & Mellish, C. (2004). Modelling politeness in natural language generation. In *Proc. of the International Natural Language Generation Conference*.

Prasad, R., Joshi, A., Dinesh, N., Lee, A., & Miltsakaki, E. (2005). The Penn Discourse TreeBank as a resource for natural language generation. In *Proc. of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*.

Prevost, S. (1995). *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation*. Ph.D. thesis, University of Pennsylvania.

Prince, E. F. (1985). Fancy syntax and shared knowledge. *Journal of Pragmatics*, *9*(1), 65–81.

Rambow, O., Rogati, M., & Walker, M. (2001). Evaluating a trainable sentence planner for a spoken dialogue travel system. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Rambow, O., & Korelsky, T. (1992). Applied text generation. In *Proc. of the Conference on Applied Natural Language Processing*.

Reiter, E. (2002). Should corpora be gold standards for NLG?. In *Proc. of the International Natural Language Generation Conference*.

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

Reiter, E., & Sripada, S. (2002). Human variation and lexical choice. *Computational Linguistics*, *28*, 545–553.

Reiter, E., Sripada, S., & Robertson, R. (2003). Acquiring correct knowledge for natural language generation. *Journal of Artificial Intelligence Research*, *18*, 491–516.

Reitter, D., Keller, F., & Moore, J. D. (2006). Computational modeling of structural priming in dialogue. In *Proc. of the Joint Conference on Human Language Technologies and Meeting of the North American Chapter of the Association for Computational Linguistics*.

Rich, E. (1979). User modelling via stereotypes. *Cognitive Science*, *3*, 329–354.

Scott, D. R., & de Souza, C. S. (1990). Getting the message across in RST-based text generation. In Dale, R., Mellish, C., & Zock, M. (Eds.), *Current Research in Natural Language Generation*, pp. 47–73. Academic Press.

Snyder, B., & Barzilay, R. (2007). Database-text alignment via structured multilabel classification. In *Proc. of the International Joint Conference on Artificial Intelligence*.

Stenchikova, S., & Stent, A. (2007). Measuring adaptation between dialogs. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*.

Stent, A., & Guo, H. (2005). A new data-driven approach for multimedia presentation generation. In *Proc. EuroIMSA*.

Stent, A., Prasad, R., & Walker, M. (2004). Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Stent, A., Walker, M., Whittaker, S., & Maloor, P. (2002). User-tailored generation for spoken dialogue: An experiment. In *Proc. of the International Conference on Spoken Language Processing*.

Wahlster, W., & Kobsa, A. (1989). User models in dialogue systems. In *User Models in Dialogue Systems*, pp. 4–34. Springer Verlag, Berlin.

Walker, M., Rambow, O., & Rogati, M. (2002). Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language: Special Issue on Spoken Language Generation*, *16*(3-4), 409–433.

Walker, M. A., Cahn, J. E., & Whittaker, S. J. (1997). Improvising linguistic style: Social and affective bases for agent personality. In *Proc. of the First Conference on Automous Agents*.

Walker, M. A., et al. (2002). DARPA communicator: Cross-system results for the 2001 evaluation. In *Proc. of the International Spoken Language Processing Conference*.

Walker, M. A., et al. (2004). Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, *28*(5), 811–840.

Webber, B., Knott, A., Stone, M., & Joshi, A. (1999). What are little trees made of?: A structural and presuppositional account using lexicalized tag. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Yeh, C.-L., & Mellish, C. (1997). An empirical study on the generation of anaphora in Chinese. *Computational Linguistics*, *23-1*, 169–190.

Zukerman, I., & Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, *11*(1-2), 129–158.