AN ANALYSIS OF AUTOMATIC CONTENT SELECTION ALGORITHMS FOR SPOKEN DIALOGUE SYSTEM SUMMARIES

Joseph Polifroni and Marilyn Walker

The Department of Computer Science
University of Sheffield
Sheffield, S1 4DP
UNITED KINGDOM
j.polifroni@dcs.shef.ac.uk/m.a.walker@sheffield.ac.uk

ABSTRACT

Previous work on information presentation in dialogue systems has argued that a user model is essential for selecting utterance content. Other work claims that the ability to browse the data is critical for supporting information-seeking behaviors in information retrieval applications, but does not specify *how* to provide browsing support. In this work, we test the hypothesis that automatically constructed summaries based on the current dialogue state and provided incrementally during the dialogue can support users' browsing activities. We examine the impact of three factors on summary efficacy: (1) the number and type of attributes selected for summary construction; (2) the use of a decision theoretic user model; and (3) the use of association rules derived automatically from the database subset currently in-focus. Our experimental results show that equally effective summaries can be constructed using either a combination of a user model with association rules, or the "Refiner" method described in previous work.

***Index Terms*—** natural language interfaces, user modeling

1. INTRODUCTION

One of the most challenging problems in spoken dialogue system (SDS) design is the development of general algorithms for information presentation. This challenge arises from the fact that, in most dialogue states, the system can only provide a description of a subset of the information in-focus. Furthermore, this subset must be selected and presented using algorithms that balance the user's cognitive limitations on processing information, against the user's need for enough high quality information to make progress on the task. In this paper, we explore the hypothesis that *summaries* of the subset of the database that is currently in-focus are a critical dialogue strategy for information presentation in SDSs.

Previous work on information-seeking dialogue has claimed that the ability to summarize the in-focus subset of the database is critical for information retrieval applications [2, 3]. Belkin *et al.* [1] claim that, during an information-seeking session, users may *scan* for interesting items in the database, *learn* about the structure of the database, retrieve items by *recognition* rather than by *specification*, and use *meta-level* information, such as might be provided by summaries of the data. Kamm *et al.* suggest that SDS dialogue strategies such as summaries are a verbal equivalent of the visual scanning behavior that makes graphical user interfaces effective [2]. The ELVIS system (EmaiL Voice Interactive System) provided summaries of the messages in the activated email folder whenever multiple messages matched the user's selection criteria [4]. Experiments with ELVIS evaluated three summarization strategies that varied the database attributes used to construct the summary, and showed that a summary strategy that used the dialogue context to vary the selected attributes was the most effective. However these experiments assumed that summaries would be useful rather than directly testing that hypothesis.

Other work claims that a user model is essential for selecting utterance content [7]. Walker *et al*[5] utilized decision-theoretic user models in an SDS for restaurant information to select a subset of the database entities in-focus in the current dialogue state, and then to select a subset of the attributes of those entities to use in a summary, recommendation, or comparison. An evaluation of the impact of the user model showed that recommendations and comparisons based on a user model were more effective, but that the user model had no impact on the effectiveness of summaries. However, these experiments did not systematically vary methods for constructing summaries.

Polifroni *et al.*[6] present a "Refiner" algorithm for summary construction that identifies attributes whose top four values account for 80% of the currently in-focus subset of the database. The dialogue in Table 1 illustrates the Refiner algorithm. They claim that summary responses, such as those in Table 1, help the user mine through large datasets to a few well-informed choices. However, they did not evaluate this claim through user experiments.

User:	Tell me about restaurants in London.
System:	I know of 596 restaurants in London. All price ranges are represented. Some of the cuisine options are Italian, British, European, and French.
User:	I'm interested in Chinese food.
System:	I know of 27 restaurants in London that serve Chinese cuisine. All price ranges are represented. Some are near the Leicester Square tube station.
User:	How about a cheap one?
System:	I know of 14 inexpensive restaurants that serve Chinese cuisine. Some are near the Leicester Square tube station. Some are in Soho.

Table 1. Example dialogue using Refiner method for summary construction.

Thus, while previous work suggests that summaries should be useful in SDSs, no one has both systematically varied methods for summary construction and evaluated their effectiveness. In this paper, we explore different methods for summary construction and test the hypothesis that automatically constructed summaries, based on the current dialogue state, and provided incrementally during the dialogue, provide effective support for users' information-seeking activity. We examine the impact of three factors on summary efficacy: (1) the type and number of attributes to be used in the summary; (2) the use of association rules derived automatically from the set of database entities in-focus to generate clusters of these entities; and (3) the use of a decision theoretic user model for attribute selection and cluster scoring. Our experimental results show that equally effective summaries can be constructed using either a combination of a user model with association rules, or a modification of the Refiner method.

2. CONSTRUCTING SUMMARIES

One of the challenging aspects of evaluating the utility of summaries in SDSs is the vast number of potential algorithms for summary construction. We assume that any summary construction algorithm takes as input a set of database entities specified in terms of attributes and their values. The restaurant database used here consists of 596 individual restaurants in London, with up to 19 attributes and their values; the examples and experiments utilize the attributes *cuisine*, *neighborhood*, *food quality*, *price*, *service quality* and *subway stop*.

The set of database entities that are input to the summary construction algorithm in each dialogue state consist of the subset of the database that is *in-focus* in the current dialogue state. At the beginning of the dialogue, the in-focus set consists of the whole database. Once the user has specified values for particular attributes, e.g. *Chinese* food, then the in-focus subset consists of restaurants that match this criterion.

As mentioned above, one parameter of summary construction is the type and number of attributes to be used in the summary. What is needed is an attribute selection algorithm that produces summaries that account for relatively large numbers of database instances and that selects attributes with values worth speaking about. Given the fact that there are many more attributes per database entity than can possibly be used in a summary, we characterize the problem of selecting which attributes to use in terms of algorithms for **Attribute Ranking** discussed below. Because the in-focus set is typically too large to be described in full, a second parameter explores techniques for using the selected attributes to produce clusters of the in-focus set to speak about (**Subset Clustering** below). Another parameter in summary construction is how many of the top-ranked attributes to use in a summary (**Number of Attributes**). The final parameter is the method for scoring the clusters derived by the clustering algorithm (**Cluster Scoring**). Finally, the algorithms used are described under **Algorithms for Constructing Summaries** below.

Attribute Ranking

We explore two candidates for attribute ranking. One algorithm is the ranking method used in decision-theoretic user models to provide an attribute ranking specific to each user [7, 3]. While previous work was unable to show that user models had any impact on the efficacy of summaries [3], this may have been an artifact of the summary construction method.

The other attribute ranking algorithm is derived from the Refiner algorithm for summary construction [6]. The original Refiner algorithm chose attributes whose top 4 unique values accounted for 80% of the total for that attribute. For example, 82% of Indian restaurants in the London database were in the neighborhoods Mayfair, Soho, and Chelsea. *Neighborhood* was, therefore, chosen as an attribute to speak about for Indian restaurants. The thresholds (i.e. 80%) were set *a priori*, so it was possible that no attribute met or exceeded the thresholds for a particular subset of the data, resulting in no summary response. In addition, every attribute in our database is instantiated with a particular value, if known (e.g., *Chinese* cuisine, neighborhood *Chelsea*), or *UNK* for values that are unknown. Unknown values are particularly common for attributes that require a rating, such as decor or food quality, so the value *UNK* can account for a large percentage of the tuples for a particular attribute. To insure that all user queries resulted in some summary responses and to avoid summaries created using attributes that are unknown (e.g., "Many of the restaurants have an unknown rating for decor."), we modified the Refiner method to utilize a ranking function. The ranking is derived by subtracting the percentage of unknown values contained in the top 4 values from the overall percentage of attributes. The resulting score is then used to rank all the attributes.

Subset Clustering

Because the in-focus set is typically too large to be described in full, a second parameter is used to determine how the attributes are used to produce clusters of the in-focus set. The simplest algorithm for producing clusters utilizes a specified number of the top-ranked attributes to define a cluster corresponding to each attribute, as in the Refiner examples in Table 1. Single attributes typically yield relatively large clusters, for example *There are 59 Chinese restaurants*.

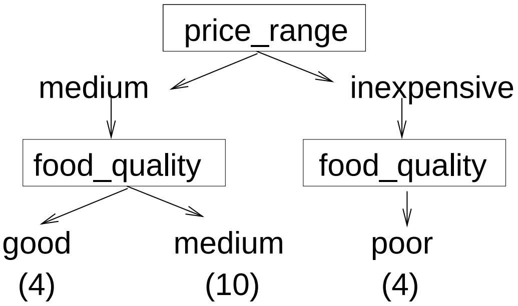

Fig. 1. A partial tree for Indian restaurants in London, using price range as the predictor variable and food quality as the dependent variable. The numbers in parentheses indicate the size of the cluster of restaurants described by the path from the root.

However, we hypothesize that more informative and useful clusters might be discovered through the use of data-mining techniques that identify *associations* between attributes. For example clusters consisting of associations between the attributes *price* and *cuisine* could produce utterances such as *There are 49 medium-priced restaurants that serve Italian cuisine.* Decision tree induction has been shown to be an efficient way to compute association rules among attributes [8]. Thus, we implemented a mechanism for generating clusters using tree induction to discover association rules about the data in-focus at each turn in the dialogue. When building a decision tree, a particular attribute is designated as the dependent variable, with other attributes used to predict that variable. Each branch in the tree represents a cluster described by the attribute/value pairs that predict the leaf node. Figure 1

No. of attributes used	Avg. bin size
2	60.4
3	4.22
4	2.23
5	2.16

Table 2. Influence of number of attributes used to construct tree and average size of the resulting leaf clusters. The numbers are averaged over trees built using the entire set of 596 restaurants in our dataset.

shows a partial tree induced from the set of Indian restaurants in the London database.

Number of Attributes

One of the most important evaluation criteria is the size of the clusters at the leaf node. An important consideration in cluster size is the number of attributes used in tree induction. Table 2 shows the influence of number of attributes on resulting cluster sizes. When using just two attributes, the average cluster size at the leaf node is 60.4. This number drops precipitously when three attributes are used. We decided to use two attributes with tree induction, to insure that the resulting clusters represented large enough subsets of our data to be of interest for summarization.

Cluster Scoring

The final parameter is the method used to score the clusters. One scoring metric is the size of the clusters produced. Single attributes typically produce large clusters, while association rules produce smaller clusters. An alternative scoring method is to use the user model to score clusters by selecting clusters that represent attribute values predicted to be of high utility for individual users (see [3] for a more detailed description of the algorithm for using the user model as a scoring function). While in principle, we could use cluster size scoring with user model attribute ranking, here we consistently use cluster size scoring with Refiner ranking and user model scoring with user model attribute ranking. A typical summary statement uses the top 3 clusters from tree induction. When not using tree induction, selecting the top 3 attribute clusters produces summaries that are of parallel length to those that use tree induction.

Algorithms for Constructing Summaries

In sum, the algorithm for constructing summary responses is:

- Take as input the subset of database entities currently in-focus;
- Rank attributes relevant to current state, using Refiner or user model ranking;
- Select top-N attributes for use in response, where $N = 2$ if we are using tree induction and $N = 3$ if we are using single attributes;
- Use single attributes for clustering or use tree induction to find clusters via association rules;
- Rank clusters with user model, if using user model; otherwise, score clusters by cluster size;
- Construct frames for generation, perform aggregation on equivalent predicates and generate responses.

Thus, the algorithms for attribute selection and cluster generation and scoring yield four test conditions. Table 3 shows an example of each of the four types of summaries, for the dataset containing restaurants that serve Indian cuisine in London. Summary S1 is constructed using (1) the Refiner attribute ranking; (2) no association rules; and (3) the top-3 values for each attribute. The specific quantifier to use (e.g., *some*, *many*) is determined based on the percentage of the data that is accounted for by the values mentioned. Summary S2 is constructed using (1) the Refiner attribute ranking; (2) tree-induction for clustering, and (3) cluster size as the cluster scoring function. Summary S3 is constructed using (1) a user model with ranking as above; (2) no association rules; and (3) user model preferences to determine which values to speak about. Summary S4 is constructed using (1) a user model with ranking of price, food, cuisine, location, service, and decor; (2) tree-induction for clustering (using rules shown in Figure 1), and (3) user model scoring as the cluster scoring function.

3. EXPERIMENTAL METHOD

Experimental subjects were students not involved in the experiment and unaware of the experimental hypotheses. We collected user model data from the 15 subjects using the procedure in [3]. The user models were then used to construct four summary statements for eight tasks in the London restaurant domain. These tasks were selected to utilize a range of attributes in the database. Four of the tasks (*large set tasks*) led to in-focus subsets of the database larger than 100 entities and four of the tasks resulted in in-focus subsets smaller than 100 entities (*small set tasks*).

Each task was presented to the subject on its own web page with the four potential system responses representing the four experimental conditions presented as text on the web page. Each subject was asked to carefully read and rate each alternative summary response on a Likert scale of $1 \ldots 5$ in response to the statement, *This response contains information I would find useful when choosing a restaurant.* The subjects were also asked to indicate which response they considered the best response and which response was the worse, and were given the opportunity to provide free-text comments about each response.

4. EXPERIMENTAL RESULTS

We performed an analysis of variance with attribute ranking method (user model vs. refiner), clustering method (association rules vs. single attributes), and set size (large vs. small) as the independent variables and user score as the independent variable. There was a main effect for set size (df = 1, F = 6.7, $P < .01$), with summaries describing small datasets getting higher scores (3.3 average score against 3.1 for summaries describing large datasets).

There was also a significant interaction between attribute ranking method and clustering method (df = 1, F = 26.8, $P <$ .00001). The set of scores at the top of Table 5 show the average user scores for each of the four possible summary types. The two highest scoring summary types used (1) no association rules and no user model ranking (average score: 3.4) and (2) association rules with user model ranking (average score:

ID	Ranking method	# of attributes	Clustering method	Cluster scoring	Summary
S1	Refiner	3	Single value	Cluster size	*I know of 35 restaurants in London serving Indian food. All price ranges are represented. Some of the neighborhoods represented are Mayfair, Soho, and Chelsea. Some of the nearby tube stations are Green Park, South Kensington and Piccadilly Circus.*
S2	Refiner	2	Associative	Cluster size	*I know of 35 restaurants in London serving Indian food. There are 3 medium-priced restaurants in Mayfair and 3 inexpensive ones in Soho. There are also 2 expensive ones in Chelsea.*
S3	User model	3	Single value	User model	*I know of 35 restaurants in London serving Indian food. There are 6 with good food quality. There are also 12 inexpensive restaurants and 4 with good service quality.*
S4	User model	2	Associative	User model	*I know of 35 restaurants in London serving Indian food. There are 4 medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality.*

Table 3. Example summaries for the four different experimental conditions.

3.4). A paired *t*-test showed no statistical difference between these two scores.

We examined optionally-provided user comments for both of these top-ranked summary types. When users preferred the Refiner method with refiner ranking, they often spoke favorably of information presented “in a general way” and liked that the responses took into account neighborhood. Since neighborhood was ranked among the top three preferences only once among all the user models collected, the preference for hearing about neighborhood appears to be unrelated to the user model. When users preferred responses constructed using association rules and user model ranking, they spoke favorably of the “overview” nature of the information presented and of how the responses highlighted “trade offs” among attributes. Thus associations among attributes are important to users, but only when those attributes are tailored to their preferences.

In addition, there was a significant interaction between attribute ranking method and setsize (df = 1, F = 11.7, $P <$ .001). The set of scores at the bottom of Table 5 show this interaction. While overall users rank summaries constructed for small datasets higher, if a user model is used, users give higher ratings to summaries for large datasets. With small datasets, users preferred summaries that did not utilize user model information.

5. CONCLUSIONS/FUTURE WORK

Automating the process of constructing summaries for SDSs depends on a selecting attributes to speak about and clustering methods over those attributes to discover what to say. We examine several methods for attribute selection and summary construction. Our experiments suggest users find that summary responses useful, and that there are several equally effective methods to construct them. We believe that the methods described here are domain-independent, but we plan to test these algorithms in other domains in future work. In addition, we hypothesize that providing summaries incrementally over the course of the dialogue will lead to reduced task durations and increased task success. However, the experimental paradigm here does not allow us to test this hypothesis. In future work, we plan to test these algorithms in a live dialogue system to assess their impact on other evaluation metrics such as task duration and task success.

	User model	Refiner
Association rules	3.4	2.9
Single attributes	3.0	3.4

	User model	Refiner
Small dataset	3.1	3.4
Large dataset	3.2	2.9

Table 4. Using scores showing the interaction between clustering method, attribute ranking, and dataset size in summary statements.

6. REFERENCES

[1] N.J. Belkin, C. Cool, A. Stein, and U. Thiel, “Cases, scripts, and information seeking strategies: On the design of interactive information retrieval systems,” *Expert Systems and Applications*, vol. 9, no. 3, pp. 379–395, 1995.

[2] C. A. Kamm, M. Walker, and L. R. Rabiner, “The role of speech processing in human-computer intelligent communication,” *Speech Communication*, vol. 23, pp. 263–278, 1997.

[3] M. Walker, S. J. Whittaker, A. Stent, P. Maloor, J. D. Moore, M. Johnston, and G. Vasireddy, “Speech-plans: Generating evaluative responses in spoken dialogue,” in *Proc. INLG02*, 2002.

[4] M. Walker, “An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email,” *Journal of Artificial Intelligence Research, JAIR*, vol. 12, pp. 387–416, 2000.

[5] M. Walker, S.J. Whitaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy, “Generation and evaluation of user tailored responses in multimodal dialogue,” *Cognitive Science*, vol. 28, pp. 811–840, 2004.

[6] J. Polifroni, G. Chung, and S. Seneff, “Towards the automatic generation of mixed-initiative dialogue systems from web content,” in *Proc. Eurospeech*, Geneva, 2003, pp. 2721–2724.

[7] G. Carenini and J. Moore, “A strategy for evaluating generative arguments,” in *Proc. First Int’l Conference on Natural Language Generation*, 2001, pp. 1307–1314.

[8] M. Kamber, L. Winstone, W. Gong, S. Cheng, and J Han, “Generalization and decision tree induction: efficient classification in data mining,” in *Proc. 7th International Workshop on Research Issues in Data Engineering (RIDE ’97)*, 1997, pp. 111–121.