# Utilizing Review Summarization in a Spoken Recommendation System

**Jingjing Liu, Stephanie Seneff and Victor Zue**
MIT Computer Science & Artificial Intelligence Laboratory
Cambridge, U.S.A.
{jingl,seneff,zue}@csail.mit.edu

## Abstract

In this paper we present a framework for spoken recommendation systems. To provide reliable recommendations to users, we incorporate a review summarization technique which extracts informative opinion summaries from grass-roots users' reviews. The dialogue system then utilizes these review summaries to support both *quality-based* opinion inquiry and *feature-specific* entity search. We propose a probabilistic language generation approach to automatically creating recommendations in spoken natural language from the text-based opinion summaries. A user study in the restaurant domain shows that the proposed approaches can effectively generate reliable and helpful recommendations in human-computer conversations.

## 1 Introduction

Spoken dialogue systems are presently available for many purposes, such as flight reservations (Seneff and Polifroni, 2000), telephone calls routing (Gorin et al., 1997), and subway lookup (Johnston et al., 2002). Recently, we have been exploring a next generation of intelligent dialogue systems, which can behave like a human agent and provide proactive assistance and selective recommendations (e.g., highly-rated restaurants or hotels) to users.

To enhance dialogue systems with intelligent services, we have to let the system "grow" reliable knowledge and intelligence. Luckily, there has recently been an explosive growth in the availability of public review sites (e.g., yelp.com, tripadvisor.com, etc.) which make a perfect resource for gathering collective opinions. In this paper, we will explore how to utilize summaries of public reviews to automatically generate recommendation-type conversations in spoken dialogue systems. An example of a conversation with our recommendation system is shown in Figure 1.

U: Are there any American restaurants in Brighton?
S: There are 61 American restaurants in Brighton. All of the rated restaurants have *good or excellent ambiance*. Many are on Washington Street, Commonwealth Avenue and Market Street.
U: Show me the ones that have *nice ambiance*.
S: I found 4 entries. Devlin's Restaurant and Bar *is perfect for romantic date*, Stockyard Restaurant has *friendly atmosphere*, Tasca Tapas Restaurant *has excellent ambiance*, and Wings Express *has friendly vibes*.
U: Which one has *good martinis*?
S: Devlin's Restaurant and Bar serves American food. It has *wonderful martinis*, good wine, great fish. It is *perfect for romantic date*. *Great place*.

Figure 1. A real conversation with our recommendation dialogue system in the restaurant domain ('U' is the user and 'S' is the system).

## 2 Dialogue Management

In our previous work (Liu and Seneff, 2009; Liu et al., 2010) we proposed an approach to extracting representative phrases and creating aspect ratings from public reviews. An example of an enhanced database entry in the restaurant domain is shown in Figure 2. Here, we use these "summary lists" (e.g., ":food", ":atmosphere") as well as aspect ratings (e.g., ":food_rating") to address two types of recommendation inquires: "feature-specific" (e.g., asking for a restaurant that serves good martinis or authentic seafood spaghetti), and "quality-based" (e.g., looking for restaurants with good food quality or nice ambiance).

```
{q restaurant
   :name "devlin's restaurant and bar"
   :atmosphere ("romantic date" "elegant decor")
   :place ("great place")
   :food ("wonderful martinis" "good wine" "great fish")
   :atmosphere_rating "4.2"
   :place_rating "4.2"
   :food_rating "4.3"
   :specialty ("martinis" "wine" "fish")    }
```

Figure 2. A database entry in our system.

## 2.1 Feature-specific Entity Search

To allow the system to identify feature-related topics in users' queries, we modify the context-free grammar in our linguistic parser by including feature-specific topics (e.g., nouns in the summary lists) as a word class. When a feature-specific query utterance is submitted by a user (as exemplified in Figure 3), our linguistic parser will generate a hierarchical structure for the utterance, which encodes the syntactic and semantic structure of the utterance and, especially, identifies the feature-related topics. A feature-specific key-value pair (e.g., "specialty: martinis") is then created from the hierarchical parsing structure, with which the system can filter the database and retrieve the entities that satisfy the constraints.

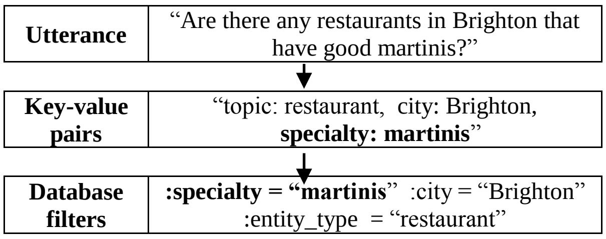

Utterance
"Are there any restaurants in Brighton that have good martinis?"
Key-value pairs
"topic: restaurant, city: Brighton, **specialty: martinis**"
Database filters
**:specialty = "martinis"** :city = "Brighton" :entity_type = "restaurant"

Figure 3. Procedure of feature-specific search.

## 2.2 Quality-based Entity Search

For quality-based questions, however, similar keyword search is problematic, as the quality of entities has variants of expressions. The assessment of different degrees of sentiment in various expressional words is very subjective, which makes the quality-based search a hard problem.

To identify the strength of sentiment in quality-based queries, a promising solution is to map textual expressions to scalable numerical scores. In previous work (Liu and Seneff, 2009), we proposed a method for calculating a sentiment score for each opinion-expressing adjective or adverb (e.g., 'bad': 1.5, 'good': 3.5, 'great': 4.0, on a scale of 1 to 5). Here, we make use of these sentiment scores and convert the original key-value pair to numerical values (e.g., "great food" → "food_rating: 4.0" as exemplified in Figure 4). In this way, the sentiment expressions can be easily converted to scalable numerical key-value pairs, which will be used for filtering the database by "aspect ratings" of entities. As exemplified in Figure 4, all the entities in the required range of aspect rating (i.e., ":food_rating ≥ 4.0") can be retrieved (e.g., the entity in Figure 2 with "food_rating = 4.3").

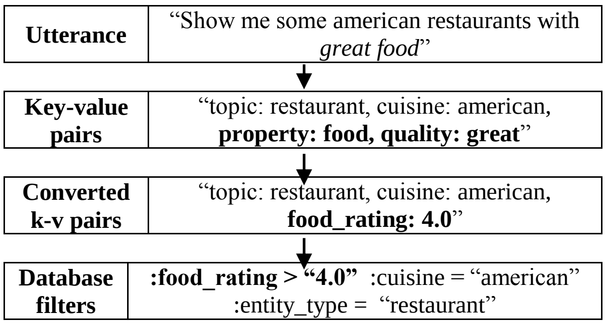

Utterance
"Show me some american restaurants with *great food*"
Key-value pairs
"topic: restaurant, cuisine: american, **property: food, quality: great**"
Converted k-v pairs
"topic: restaurant, cuisine: american, **food_rating: 4.0**"
Database filters
**:food_rating > "4.0"** :cuisine = "american" :entity_type = "restaurant"

Figure 4. Procedure of qualitative entity search.

## 3 Probabilistic Language Generation

After corresponding entities are retrieved from the database based on the user's query, the language generation component will create recommendations by expanding the summary lists of the retrieved database entries into natural language utterances.

Most spoken dialogue systems use predefined templates to generate responses. However, manually defining templates for each specific linguistic pattern is tedious and non-scalable. For example, given a restaurant with "nice jazz music, best breakfast spot, great vibes", three templates have to be edited for three different topics (e.g., "<restaurant> *plays* <adjective> music"; "<restaurant> *is* <adjective> breakfast spot"; "<restaurant> *has* <adjective> vibes"). To avoid the human effort involved in the task, corpus-based approaches (Oh and Rudnicky, 2000; Rambow et al., 2001) have been developed for more efficient language generation. In this paper, we propose a corpus-based probabilistic approach which can automatically learn the linguistic patterns (e.g., predicate-topic relationships) from a corpus and generate natural sentences by probabilistically selecting the best-matching pattern for each topic.

The proposed approach consists of three stages: 1) plant seed topics in the context-free grammar; 2) identify semantic structures associated with the seeds; 3) extract association pairs of linguistic patterns and the seeds, and calculate the probability of each association pair.

First, we extract all the nouns and noun phrases that occur in the review summaries as the seeds. As aforementioned, our context-free grammar can parse each sentence into a hierarchical structure. We modify the grammar such that, when parsing a sentence which contains one of these seed topics, the parser can identify the seed as an "active" topic (e.g., "vibes", "jazz music", and "breakfast spot").

The second stage is to automatically identify all the linguistic patterns associated with each seed. To do so, we use a large corpus as the resource pool and parse each sentence in the corpus for linguistic analysis. We modify our parser such that, in a preprocessing step, the predicate and clause structures that are semantically related to the seeds will be assigned with identifiable tags. For example, if the subject or the complement of the clause (or the object of the predicate) is an "active" topic (i.e., a seed), an "active" tag will be automatically assigned to the clause (or the predicate). In this way, when examining syntactic hierarchy of each sentence in the corpus, the system can encode all the linguistic patterns of clauses or predicate-topic relationships associated with the seeds with "active" tags.

Based on these tags, association pairs of "active" linguistic patterns and "active" topics can be extracted automatically. For each seed topic, we calculate the probability of its co-occurrence with each of its associated patterns by:

$$prob\left(pattern_j \middle| seed_k\right) = \frac{count(pattern_j,\ seed_k)}{\sum_i count(pattern_i,\ seed_k)} \quad (1)$$

where $seed_k$ is a seed topic, and $pattern_i$ is every linguistic pattern associated with $seed_k$. The probability of $pattern_j$ for $seed_k$ is the percentage of the co-occurrences of $pattern_j$ and $seed_k$ among all the occurrences of $seed_k$ in the corpus. This is similar to a bigram language model. A major difference is that the linguistic pattern is not necessarily the word adjacent to the seed. It can be a long distance from the seed with strong semantic dependencies, and it can be a semantic chunk of multiple words. The long distance semantic relationships are captured by our linguistic parser and its hierarchical encoding structure; thus, it is more reliable than pure co-occurrence statistics or bigrams. Figure 5 shows some probabilities learned from a review corpus. For example, "is" has the highest probability (0.57) among all the predicates that co-occur with "breakfast spot"; while "have" is the best-match for "jazz music".

| **Association pair** | **Constituent** | **Prob.** |
|---|---|---|
| "at" : "breakfast spot" | PP | 0.07 |
| "is" : "breakfast spot" | Clause | **0.57** |
| "for" : "breakfast spot" | PP | 0.14 |
| "love" : "jazz music" | VP | 0.08 |
| "have" : "jazz music" | VP | **0.23** |
| "enjoy": "jazz music" | VP | 0.08 |

Figure 5. Partial table of probabilities of association pairs (VP: verb phrase; PP: preposition phrase).

Given these probabilities, we can define pattern selection algorithms (e.g., always select the pattern with the highest probability for each topic; or rotates among different patterns from high to low probabilities), and generate response utterances based on the selected patterns. The only domain-dependent part of this approach is the selection of the seeds. The other steps all depend on generic linguistic structures and are domain-independent. Thus, this probabilistic method can be easily applied to generic domains for customizing language generation.

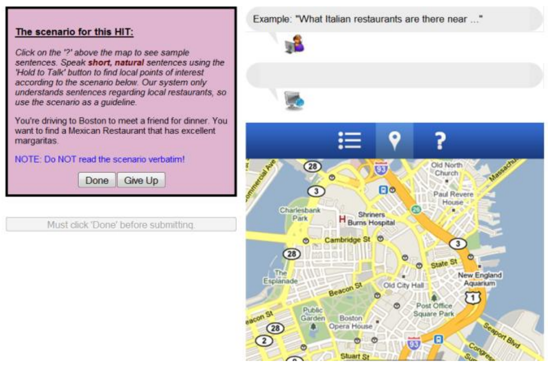
## 4 Experiments

A web-based multimodal spoken dialogue system, CityBrowser (Gruenstein and Seneff, 2007), developed in our group, can provide users with information about various landmarks such as the address of a museum, or the opening hours of a restaurant. To evaluate our proposed approaches, we enhanced the system with a review-summary database generated from a review corpus that we harvested from a review publishing web site (www.citysearch.com), which contains 137,569 reviews on 24,043 restaurants.

We utilize the platform of Amazon Mechanical Turk (AMT) to conduct a series of user studies. To understand what types of queries the system might potentially be handling, we first conducted an AMT task by collecting restaurant inquiries from general users. Through this AMT task, 250 sentences were collected and a set of generic templates encoding the language patterns of these sentences was carefully extracted. Then 10,000 sentences were automatically created from these templates for language model training for the speech recognizer.

To evaluate the quality of recommendations, we presented the system to real users via customized AMT API (McGraw et al., 2010) and gave each subject a set of assignments to fulfill. Each assignment is a scenario of finding a particular restaurant, as shown in Figure 6. The user can talk to the system via a microphone and ask for restaurant recommendations.

We also gave each user a questionnaire for a subjective evaluation and asked them to rate the system on different aspects. Through this AMT task we collected 58 sessions containing 270 utterances (4.6 utterances per session on average) and 34 surveys. The length of the utterances varies significantly, from "Thank you" to "Restaurants along Brattle Street in Cambridge with nice cocktails." The average number of words per utterance is 5.3.



**The scenario for this HIT:**

*Click on the '?' above the map to see sample sentences. Speak **short, natural** sentences using the 'Hold to Talk' button to find local points of interest according to the scenario below. Our system only understands sentences regarding local restaurants, so use the scenario as a guideline.*

You're driving to Boston to meet a friend for dinner. You want to find a Mexican Restaurant that has excellent margaritas.

NOTE: Do NOT read the scenario verbatim!

Done | Give Up

Must click 'Done' before submitting.

Example: "What Italian restaurants are there near ..."

Figure 6. Interface of our system in an AMT assignment.

Among all the 58 sessions, 51 were successfully fulfilled, i.e., in 87.9% of the cases the system provided helpful recommendations upon the user's request and the user was satisfied with the recommendations. Among those seven failed cases, one was due to loud background noise, two were due to users' operation errors (e.g., clicking "DONE" before finishing the scenario), and four were due to recognition performance.

The user ratings in the 34 questionnaires are shown in Figure 7. On a scale of 0 (the center) to 5 (the edge), the average rating is 3.6 on the easiness of the system, 4.4 on the helpfulness of the recommendations, and 4.1 on the naturalness of the system response. These numbers indicate that the system is very helpful at providing recommendation upon users' inquiries, and the response from the system is present in a natural way that people could easily understand.

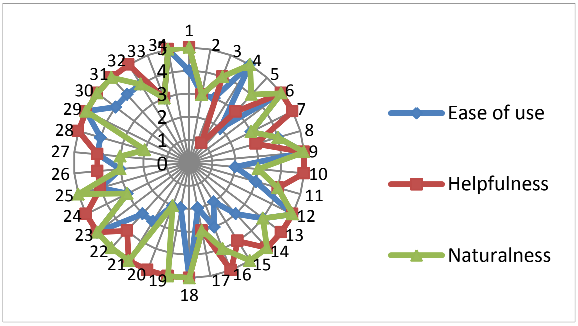


Figure 7. Users' ratings from the questionnaires.

The lower rating of ease of use is partially due to recognition errors. For example, a user asked for "pancakes", and the system recommended "pizza places" to him. In some audio clips recorded, the background noise is relatively high. This may be due to the fact that some AMT workers work from home, where it can be noisy.

## 5 Conclusions

In this paper we present a framework for incorporating review summarization into spoken recommendation systems. We proposed a set of entity search methods as well as a probabilistic language generation approach to automatically create natural recommendations in human-computer conversations from review summaries. A user study in the restaurant domain shows that the proposed approaches can make the dialogue system provide reliable recommendations and can help general users effectively.

Future work will focus on: 1) improving the system based on users' feedback; and 2) applying the review-based approaches to dialogue systems in other domains.

## Acknowledgments

This research is supported by Quanta Computers, Inc. through the T-Party project.

## References

Gorin, A., Riccardi, G., and Wright, J. H. 1997. How May I Help You? *Speech Communications*. Vol. 23, pp. 113 – 127.

Gruenstein, A. and Seneff, S. 2007. Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms. *In Proc. the 8th SIGdial Workshop on Discourse and Dialogue*, pp. 111—119.

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. ACL*, pp. 376 – 383.

Liu, J. and Seneff, S. 2009. Review sentiment scoring via a parse-and-paraphrase paradigm, *In Proc. EMNLP*, Vol. 1.

Liu, J., Seneff, S. and Zue, V. 2010. Dialogue-Oriented Review Summary Generation for Spoken Dialogue Recommendation Systems. *In Proc. NAACL-HLT*.

McGraw, I., Lee, C., Hetherington, L., Seneff, S., Glass, J. 2010. Collecting Voices from the Cloud. *In Proc. LREC*.

Oh, A.H. and Rudnicky, A.I. 2000. Stochastic Language Generation for Spoken Dialogue Systems. *In Proc. of ANLP-NAACL*, pp. 27-32.

Rambow, O., Bangalore, S., Walker, M. 2001. Natural Language Generation in Dialog Systems. *In Proc. Human language technology research*.

Seneff, S. and Polifroni, J. 2000. Dialogue Management in the Mercury Flight Reservation System. *In Proc. Dialogue Workshop, ANLP-NAACL*.