INTERSPEECH 2007, pp. 2713-2716, ISCA, Antwerp, Belgium, 2007

The Voice-Rate Dialog System for Consumer Ratings

Geoffrey Zweig, Patrick Nguyen, Y. C. Ju, Ye-Yi Wang, Dong Yu, Alex Acero

Microsoft Research, Redmond, WA USA

{gzweig,panguyen,yuncj,yeyiwang,dongyu,alexac}@microsoft.com

Abstract

Voice-Rate is an experimental dialog system that makes product and business ratings available to consumers via a toll-free phone number. By calling Voice-Rate, users can access the ratings of more than one million products, a quarter million local businesses (restaurants), and three thousand national businesses. This paper describes the Voice Rate system, and solutions to three key technical challenges: robust name-matching, efficient disambiguation, and review synthesis for telephone playback. Voice-Rate can be accessed by calling 1-877-456-DATA (toll-free) within the U.S.

Index Terms: dialog systems, consumer ratings, speech recognition, disambiguation, review summarization

1. Introduction

In recent years, web-based rating systems have provided a valuable service to consumers by allowing them to share their assessments of goods and services, thus enabling more informed decision making. The use of these systems, however, requires access to a web interface – typically a laptop or desktop computer - and this restricts their usefulness to well-planned purchases. While mobile phones also provide some web access, their small screens make them inconvenient to use. Thus, today's consumer has no really viable source of information to draw on when he or she is at a store considering a purchase that has not been planned in advance. The same is true when someone who is out and about is considering going to a newly-found restaurant or business. Voice-Rate has been designed to fill this "information-gap" by making ratings available to cell-phone owners via a toll-free number and a dialog interface.

Voice-Rate offers ratings for three broad classes of items: products, local businesses, and national businesses. Products are fairly straightforward and include items in many of the standard categories found, for example, on MSN Shopping. It is necessary, however, to break businesses into two further subcategories: local businesses and national businesses. This distinction is necessary because many businesses such as restaurants or small stores have common names that occur in many locations. Thus the name by itself is insufficient – the "Las Margaritas" restaurant of Redmond, WA has nothing to do with the "Las Margaritas" of Cleveland, Ohio. In contrast, some businesses are more natural to discuss at the national level, for example airlines and car-rental companies. These considerations led to a three-way classification.

While the ultimate goal of Voice-Rate is to establish a sufficiently large community of users that user-supplied ratings will be sufficient in terms of the number of businesses and products covered, this is not the case for a newly deployed system. Therefore, we have primed Voice-Rate with reviews taken from MSN Shopping and similar web-based rating services. These reviews include over 1.1 million distinct products (covering a variety of household items ranging from electronics to DVDs and books), over a quarter-million local businesses in 1600 metro areas, and approximately three thousand national businesses. The local businesses are currently all restaurants; as more reviews are accumulated, this will be extended to shops and other user-determined entities.

Before proceeding, we present a sample dialog to illustrate how Voice-Rate is used. In this scenario, John is on a business trip, and has just arrived at his hotel in Allentown, PA. Across the street is a restaurant called "Two Eleven York," and while John is hungry, the restaurant is completely empty which makes him nervous. To decide, John takes out his cell-phone and calls Voice-Rate.

- **System:** Welcome to Voice-Rate. Would you like to get a rating or leave a rating?
- **John:** Get a rating.
- **System:** Do you want a rating for a local business, a national business or a product?
- **John:** A local business.
- **System:** What city and state?
- **John:** Allentown, Pennsylvania.
- **System:** Which business?
- **John:** "Two Eleven York"
- **System:** I heard you say, "Two Eleven York." Is that correct?
- **John:** Yes.
- **System:** This is rated 4.4 out of 5. Here are some of the things people had to say about this restaurant. Positive comments: "The shell crab is out of this world…" Negative comments: "The service was horrible…"

The development of Voice-Rate required solutions to three key problems. First, people rarely ask for a complete business name, or a product name as it is specified on a box. Thus, it was necessary to implement a very robust name matching procedure. Second, even when complete, names can be highly ambiguous – popular movies like Star Wars, for example, often have related products in categories such as *DVDs*, *Toys and Games*, *Music (Soundtrack)*, and so on. The possibility of speech recognition errors further compounds the problem of uniquely determining the business or product. To address this problem, we present a reliable disambiguation strategy that revolves around separating speech recognition errors from search errors and addressing them separately.

The third technical challenge is to provide a brief, crisp summary of the available reviews that is both richer than a single number and at the same time short enough to be suitable for a telephone interaction. To do this, we developed

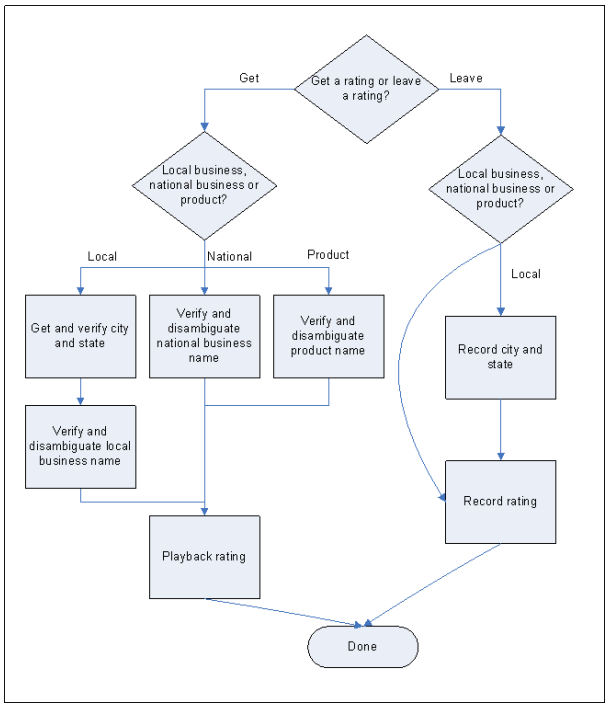

Figure 1: *High Level Voice-Rate Dialog Structure.*

a novel method for extracting short, pithy remarks from reviews, and identifying them as positive or negative. These are then played to callers and significantly richen the user experience. The extraction is based on training a maximum entropy classifier to map from text reviews to numerical ratings. The features that are used are word n-grams, and the presence of highly weighted n-grams in a punctuation-delimited section of text marks that section for inclusion in a summary.

To our knowledge, Voice-Rate is the first large scale ratings dialog system. However, the technology behind it is closely related to previous dialog systems, especially directory assistance or "411" systems, e.g. [1,2,3,4]. The field of name-matching and lookup is also well-developed, and a general discussion of robust name-matching techniques can be found in [5].

The second area of related research has to do with web rating systems. Interesting work on extracting information from such ratings can be found in, e.g. [6,7]. Work has also been done using text-based input to determine products that are relevant to a given query [8]. Our own work differs from this in that it focuses on *spoken* input, and in its *breadth* – covering both products and businesses.

The remainder of this paper is organized as follows. Section 2 describes the Voice-Rate dialog structure and provides information about the underlying speech recognition. Section 3 is devoted to the search and disambiguation procedure that is used, and Section 4 covers our review summarization process. Before concluding in Section 6, we present some quantitative evaluations of both our name-lookup search, and rating prediction algorithms in Section 5.

2. System Basics

2.1.1. Dialog Structure

The high level Voice-Rate dialog structure is shown in Figure 1. While this structure is relatively straightforward, it is worth noting that the rating-giving process is significantly different from the rating-getting process. When a user asks for a rating, it is imperative that the system correctly identifies the item to be rated. A mistake would result in the transmittal of incorrect and potentially slanderous information, and therefore the system engages in a confirmatory dialog. In contrast, the process of leaving a rating is "fail-soft," and in order to make it as fast as possible, user responses are simply recorded for offline analysis. Subsequent low confidence analyses are simply be discarded.

2.1.2. Automatic Speech Recognition

Voice-Rate does recognition with the Microsoft Speech Server (MSS), an off-the-shelf commercial recognizer. This is an HMM based system which uses 36 dimensional MFCC features, triphone cross-word acoustic context with a 6,000 leaf decision tree, and approximately 96,000 gaussians [9]. The feature vectors are formed by concatenating 12 dimensional MFCCs and energy with first, second and third derivatives, and projecting to 36 dimensions with HLDA. The vocabulary size was 190k words. Bigram language models were used; the product grammar had 206k bigrams, while the business grammars were smaller and varied in size for the 1600 localities in the system. The dialog system itself was built with MSS developer tools.

3. Search and Disambiguation

The core problem that Voice-Rate must solve is how to identify exactly which business or product a caller wants a rating for. The solution we adopt works in two stages. First, the system takes its "best-shot" and returns the item that is most similar to the ASR output, as measured by the term-frequency inverse-document-frequency (TF-IDF) metric [5]. This is then presented to the user for verification, and if it is not the correct item, a correction and disambiguation dialog begins. The following subsections discuss our use of TF-IDF and the disambiguation strategy.

3.1. TF-IDF Metric

Voice-Rate solves the fuzzy-matching problem by treating spoken queries as well as business and product names as documents, and then performing TF-IDF based lookup. Briefly, in a TF-IDF implementation, each item (product name, business name, or query) is represented as a N-dimensional vector, where N is the size of the vocabulary. For a given item i, the j^{th} dimension of this vector has the TF-IDF weight w_{ij} of the j^{th} vocabulary word with respect to the entry. This weight is given by:

$$w_{ij} = \frac{c_{ij}}{C_i} \log \frac{N}{N_j} \qquad (1)$$

In this equation, c_{ij} is the count of the number of times word j occurs in item i's name; C_i is the total number of words in the item's name; N is the total number of items, and N_j is the number of items that contain word j. Distance between two items is defined as the cosine of the angle between their respective vectors, which can be computed with a simple (normalized) dot-product.

N-gram	Snippet
worst pizza	I had some of the worst pizza I've ever been served outside of a microwave box
was excited	I had read all of the wonderful reviews and was excited to try it
worst restaurant	absolutely the worst restaurant I've been to in years
courteous	courteous and friendly owners
worst food	was the worst food experience I have had in years in SF

Table 1: *The five most heavily weighted n-grams (left) and sample snippets that contain them (right).*

The TF-IDF metric has a number of useful properties. First, it does not weigh all words equally. Common words like "Pizzeria" will receive a low IDF score, while words that are infrequent (and thus informative) like "Abbondanza" will receive high scores. Second, it makes no reference to word order, and is thus robust to the reordering that people may introduce (e.g. "Samsung SyncMaster 214T" vs. "Samsung 214T SyncMaster").

3.2. Disambiguation Strategy

In the ideal case, after a user asks for a particular product or business, the best-matching item as measured by TF-IDF would be the one intended by the user. In reality, of course, this is often not the case, and further dialog is necessary to determine the user's intent. In the case of business names, after an error we simply ask the user to repeat the request. For product disambiguation, however, we have implemented more sophisticated disambiguation strategy as described below.

When a user calls Voice-Rate and asks for a product review, the system solicits the user for the product name, does TF-IDF lookup, and presents the highest-scoring match for user confirmation. If the user does not accept the retrieved item, Voice-Rate initiates a disambiguation dialog.

Aside from inadequate product coverage, which cannot be fixed at runtime, there are two possible sources for error: automatic speech recognition (ASR) errors, and TF-IDF lookup errors. The disambiguation process begins by eliminating the first. To do this, it asks the user if his or her exact words were the recognized text, and if not to repeat the request. This loop iterates twice, and if the user's exact words still have not been identified (i.e. the ASR-output acknowledged correct by the user), Voice-Rate apologizes and hangs up.

Once the user's exact words have been validated, Voice-Rate gets a positive identification on the product category. From the set of high-scoring TF-IDF items, a list of possible categories is compiled. For example, for "The Lord of the Rings The Two Towers," there are items in *Video Games*, *DVDs*, *Music*, *VHS*, *Software*, *Books*, *Websites*, and *Toys and Games*. These categories are read to the user, who is asked to select one. All the close-matching product names in the selected category are then read to the user, until one is selected or the list is exhausted. A quantitative assessment of this process is presented in Section 5.

Query	Intended Product
Da Vinci Code	The Da Vinci Code (in Audio CDs)
Grand Theft Auto	Grand Theft Auto Vice City (in Video Games)
Learning Table	Leapstart Learning Table (in Toys and Games)

Table 2: *TF-IDF Test set examples.*

	# Matches	TF-IDF Search Accuracy
Top Level	27	0.48
Given Category	6.7	0.69

Table 3: *Number of matching items, and one-best accuracy with and without category information.*

4. Selecting Representative Snippets

In addition to having a simple numerical rating, it is often desirable to have more detailed information about what people like and dislike about the reviewed item. One simple way of providing this information would be to simply retrieve the ratings on file, and present them verbatim to a user. On a mobile phone, however, the output capabilities are limited, and it might not be possible to render all of the information available, either on the small screen, or by voice - there are sometimes dozens of reviews for a given business or product. To solve this problem, we have taken the approach of extracting *snippets* from the review to serve as a summary. (By snippet we mean a section of punctuation-delimited text.) There are two reasons for this. First, we believe that salient, specific and distinctive statements are very helpful in contrasting different items. Secondly, in contrast to a generative approach, in which one builds an ontology of information in the database, and then creates statements conveying the perceived quality, the snippets provide a direct form of community feedback. This strategy has been implemented for restaurants as described below.

There is a dual objective in selecting what review snippets are most appropriate for summarization. Firstly, snippets bearing the most pronounced opinions are preferred. To identify such sections, we built a maximum entropy model (maxent) for predicting numerical rating from textual reviews. The classifier uses 326k word-unigrams and bigrams as features, selected from our database of 306k restaurant reviews using a count-cutoff threshold. After the classifier was built, it was used to identify candidate snippets. Each punctuation-delimited sequence of words was considered, and those containing n-grams with high total maxent weight were selected. Thus, the snippets were selected according to their contribution to the entire maxent score. Table 1 shows the most heavily weighted n-grams, and a sample snippet that includes each.

The second objective in choosing snippets is to maximize coverage of all the relevant qualitative areas – for example, for restaurants: food, atmosphere, service, and value. Again, we built a maxent classifier for that task. This classifier predicts the relevance of a snippet to a category like "service," and was trained on 23k snippets that were manually labeled. Each candidate snippet is then automatically labeled according to category. The final set of snippets for a

	1/5 (poor)	2/5	3/5	4/5	5/5 (great)
Prior	9.4%	6.2%	8.9%	18%	57%
Maxent err	69%	54%	60%	51%	23%

Table 4: *Rating priors, and maxent error rate.*

restaurant is formed by taking the most positive and negative reviews for each of the categories.

5. Quantitative Measures

This section presents quantitative measures of the TF-IDF search procedure, end-to-end system performance, and of rating classification.

5.1. TF-IDF Search

5.1.1. Test Set

To measure the effectiveness of the TF-IDF based search, we used a text-based test set constructed from MSN Shopping queries. The MSN Shopping queries consist of the 100,000 most frequent queries entered into the MSN Shopping search box. The test set was formed by taking each product in the database and doing a "reverse TF-IDF lookup" to return all the queries within a given degree of similarity (cosine similarity of 0.7 or greater). The resulting queries were taken as reasonable ways to ask for the specified product. The resulting test set has just over 40,000 such query/product pairs. Several examples are given in Table 2.

5.1.2. TF-IDF Lookup Accuracy

Once the test set was defined, a TF-IDF lookup was made into the product database for each query. Table 3 shows the average number of products with a cosine similarity to the query of 0.7 or better, and the accuracy obtained with the single highest-scoring TF-IDF scoring item. Ties were broken by favoring the item with more reviews. Results are provided for both "top-level" queries in which the category is not known, and also computed conditioned on the product category. This table indicates that once the category is known, there are on average a relatively small number (6.7) of high-scoring product names which can be read to a caller.

5.2. End-to-End System Performance

To get a sense of the overall system performance, a set of fifty product names was selected at random and divided among five volunteer callers. Callers were asked to insist on the correct category as well as name in the returned result, possibly necessitating disambiguation. Callers were instructed to ask for a product in as natural a way as possible. Of the fifty calls, 73% were successful: 59% resulted in immediate success without any disambiguation necessary; a further 14% were successful after the disambiguation dialog, and 27% resulted in failure. The failed calls were almost uniformly due to speech recognition errors, sometimes caused by homonyms which the system has no way of dealing with. Interestingly, about a third of the successful calls were successful in spite of the existence of speech recognition errors; in such cases, enough informative words were decoded for TF-IDF to return the correct item despite the errors.

5.3. Rating Prediction

Our rating prediction classifier was tested on restaurant ratings. From our database of 306k reviews of 208k distinct restaurants containing a total of 24M words, 10k reviews were held out for development, and a further 10k for evaluation. Table 4 shows the prior distribution over rating values, as well as the error rate of the maxent system. It can be seen that the user ratings are skewed towards high values, with only about 15% of all restaurants getting a rating of 1 or 2. While the overall error rates are high, we have found that nevertheless the highly weighted maxent features are informative, and suitable for our purpose.

Another way to judge the system is by the mean-square error (MSE) between the predicted and true ratings. Computed from the prior, the default rating that minimizes MSE is 4. The standard deviation of that constant rating is 1.30, that is, we make an error of typically more than a point using that system. The standard deviation using the classifier was reduced to 0.67.

6. Conclusions

This paper presents a dialog system for accessing reviews of products and businesses over the phone. We find that an effective disambiguation strategy results from separating speech recognition from lookup errors, and disambiguating on product categories. Almost all the errors are due to faulty speech recognition. We further describe a novel method for selecting informative snippets from reviews for playback over the phone.

7. References

[1] C. A. Kamm, K. M. Yang, C. R. Shamieh and S. Singhal. "Speech recognition issues for directory assistance applications," Second IEEE Workshop on Interactive Voice Technology for Telecoms Applications. 1994.

[2] P. Natarajan, R. Prasad, R. Schwartz and J. Makhoul. "A Scalable Architecture for Directory Assistance Automation," ICASSP 2002.

[3] E. Levin and A. M. Manš. "Voice User Interface Design for Automated Directory Assistance," Eurospeech 2005.

[4] E. E. Jan, B. Maison, L. Mangu and G. Zweig. "Automatic construction of Unique Signatures and Confusable sets for Natural Language Directory Assistance Application," Eurospeech 2003.

[5] W. W. Cohen, P Ravikumar and S. E. Fienberg. "A comparison of string distance metrics for name-matching tasks," Proc. IJCAI-2003 Workshop on Information, 2003.

[6] M. Hu and B. Liu. "Mining and summarizing customer reviews," Proc. 2004 ACM SIGKDD Intl. Conf.

[7] M. Gamon, A. Aue, S. Corston-Oliver and E. Ringger. "Pulse: Mining Customer Opinions from Free Text." In Lecture Notes in Computer Science. Vol. 3646. Springer Verlag. (IDA 2005), pages 121-132.

[8] J. Chai, V. Horvath, N. Nicolov, M. Stys, N. Kambhatla, W. Zadrozny and P. Melville. "Natural Language Assistant - A Dialog System for Online Product Recommendation," AI Magazine (23), 2002.

[9] Yu, L. Deng, X. He, A. Acero. "Large Margin Minimum Classification Error Training for Large-Scale Speech Recognition Tasks," ICASSP 2007.