Learning about Voice Search for Spoken Dialogue Systems

Rebecca J. Passonneau[1], Susan L. Epstein[2,3], Tiziana Ligorio[2],
Joshua B. Gordon[4], Pravin Bhutada[4]

[1]Center for Computational Learning Systems, Columbia University
[2]Department of Computer Science, Hunter College of The City University of New York
[3]Department of Computer Science, The Graduate Center of The City University of New York
[4]Department of Computer Science, Columbia University
becky@cs.columbia.edu, susan.epstein@hunter.cuny.edu, tligorio@gc.cuny.edu,
joshua@cs.columbia.edu, pravin.bhutada@gmail.com

Abstract

In a Wizard-of-Oz experiment with multiple wizard subjects, each wizard viewed automated speech recognition (*ASR*) results for utterances whose interpretation is critical to task success: requests for books by title from a library database. To avoid non-understandings, the wizard directly queried the application database with the ASR hypothesis (*voice search*). To learn how to avoid misunderstandings, we investigated how wizards dealt with uncertainty in voice search results. Wizards were quite successful at selecting the correct title from query results that included a match. The most successful wizard could also tell when the query results did not contain the requested title. Our learned models of the best wizard's behavior combine features available to wizards with some that are not, such as recognition confidence and acoustic model scores.

1 Introduction

Wizard-of-Oz (*WOz*) studies have long been used for spoken dialogue system design. In a relatively new variant, a subject (the *wizard*) is presented with real or simulated automated speech recognition (*ASR*) to observe how people deal with incorrect speech recognition output (Rieser, Kruijff-Korbayová, & Lemon, 2005; Skantze, 2003; Stuttle, Williams, & Young, 2004; Williams & Young, 2003, 2004; Zollo, 1999). In these experiments, when a wizard could not interpret the ASR output (*non-understanding*), she rarely asked users to repeat themselves. Instead, the wizard found other ways to continue the task.

This paper describes an experiment that presented wizards with ASR results for utterances whose interpretation is critical to task success: requests for books from a library database, identified by title. To avoid non-understandings, wizards used *voice search* (Wang et al., 2008): they directly queried the application database with ASR output. To investigate how to avoid errors in understanding (*misunderstandings)*, we examined how wizards dealt with uncertainty in voice search results. When the voice search results included the requested title, all seven of our wizards were likely to identify it. One wizard, however, recognized far better than the others when the voice search results did not contain the requested title. The experiment employed a novel design that made it possible to include system features in models of wizard behavior. The principal result is that our learned models of the best wizard's behavior combine features that are available to wizards with some that are not, such as recognition confidence and acoustic model scores.

The next section of the paper motivates our experiment. Subsequent sections describe related work, the dialogue system and embedded wizard infrastructure, experimental design, learning methods, and results. We then discuss how to generalize from the results of our study for spoken dialogue system design. We conclude with a summary of results and their implications.

2 Motivation

Rather than investigate full dialogues, we addressed a single type of turn exchange or adjacency pair (Sacks et al., 1974): a request for a book by its

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 840–848,
Los Angeles, California, June 2010.

title. This allowed us to collect data exclusively about an utterance type critical for task success in our application domain. We hypothesized that low-level features from speech recognition, such as acoustic model fit, could independently affect voice search confidence. We therefore applied a novel approach, *embedded WOz*, in which a wizard and the system together interpret noisy ASR.

To address how to avoid misunderstandings, we investigated how wizards dealt with uncertainty in voice search returns. To illustrate what we mean by uncertainty, if we query our book title database with the ASR hypothesis:

```
ROLL DWELL
```

our voice search procedure returns, in this order:

```
CROMWELL
ROBERT LOWELL
ROAD TO WEALTH
```

The correct title appears last because of the score it is assigned by the string similarity metric we use.

Three factors motivated our use of voice search to interpret book title requests: noisy ASR, unusually long query targets, and high overlap of the vocabulary across different query types (e.g., author and title) as well as with non-query words in caller utterances (e.g., "Could you look up . . .").

First, accurate speech recognition for a real-world telephone application can be difficult to achieve, given unpredictable background noise and transmission quality. For example, the 68% word error rate (*WER*) for the fielded version of Let's Go Public! (Raux et al., 2005) far exceeded its 17% WER under controlled conditions. Our application handles library requests by telephone, and would benefit from robustness to noisy ASR.

Second, the book title field in our database differs from the typical case for spoken dialogue systems that access a relational database. Such systems include travel booking (Levin et al., 2000), bus route information (Raux et al., 2006), restaurant guides (Johnston et al., 2002; Komatani et al., 2005), weather (Zue et al., 2000) and directory services (Georgila et al., 2003). In general for these systems, a few words are sufficient to retrieve the desired attribute value, such as a neighborhood, a street, or a surname. Mean utterance length in a sample of 40,000 Let's Go Public! utterances, for example, is 2.4 words. The average book title length in our database is 5.4 words.

Finally, our dialogue system, *CheckItOut*, allows users to choose whether to request books by title, author, or catalogue number. The database represents 5028 active patrons (with real borrowing histories and preferences but fictitious personal information), 71,166 book titles and 28,031 authors. Though much smaller than a database for a directory service application (Georgila et al., 2003), this is much larger than that of many current research systems. For example, Let's Go Public! accesses a database with 70 bus routes and 1300 place names. Titles and author names contribute 50,394 words to the vocabulary, of which 57.4% occur only in titles, 32.1% only in author names, and 10.5% in both. Many book titles (e.g., *You See I Haven't Forgotten, You Never Know*) have a high potential for confusability with non-title phrases in users' book requests. Given the longer database field and the confusability of the book title language, integrating voice search is likely to have a relatively larger impact in CheckItOut.

We seek to minimize non-understandings and misunderstandings for several reasons. First, user corrections in both situations have been shown to be more poorly recognized than non-correction utterances (Litman et al., 2006). Non-understandings typically result in re-prompting the user for the same information. This often leads to hyper-articulation and concomitant degradation in recognition performance. Second, users seem to prefer systems that minimize non-understandings and misunderstandings, even at the expense of dialogue efficiency. Users of the TOOT train information spoken dialogue system preferred system-initiative to mixed- or user-initiative, and preferred explicit confirmation to implicit or no confirmation (Litman & Pan, 1999). This was true despite the fact that a mixed-initiative, implicit confirmation strategy led to fewer turns for the same task. Most of the more recent work on spoken dialogue systems focuses on mixed-initiative systems in laboratory settings. Still, recent work suggests that while mixed- or user-initiative is rated highly in usability studies, under real usage it "fails to provide [a] robust enough interface" (Turunen et al., 2006). Incorporating accurate voice search into spoken dialogue systems could lead to fewer non-understandings and fewer misunderstandings.

3 Related Work

Our approach to noisy ASR contrasts with many other information-seeking and transaction-based dialogue systems. Those systems typically perform natural language understanding on ASR output before database query with techniques that try to improve or expand ASR output. None that we know of use voice search. For one directory service application, users spell the first three letters of surnames, and then ASR results are expanded using frequently confused phones (Georgila et al., 2003). A two-pass recognition architecture added to Let's Go Public! improved concept recognition in post-confirmation user utterances (Stoyanchev & Stent, 2009). In (Komatani et al., 2005), a shallow semantic interpretation phase was followed by decision trees to classify utterances as relevant either to query type or to specific query slots, to narrow the set of possible interpretations. CheckItOut is most similar in spirit to the latter approach, but relies on the database earlier, and only for semantic interpretation, not to also guide the dialogue strategy.

Our approach to noisy ASR is inspired by previous WOz studies with real (Skantze, 2003; Zollo, 1999) or simulated ASR (Kruijff-Korbayová et al., 2005; Rieser et al., 2005; Williams & Young, 2004). Simulation makes it possible to collect dialogues without building a speech recognizer, and to control for WER. In the studies that involved task-oriented dialogues, wizards typically focused more on the task and less on resolving ASR errors (Williams & Young, 2004; Skantze, 2003; Zollo, 1999). In studies more like the information-seeking dialogues addressed here, an entirely different pattern is observed (Kruijff-Korbayová et al., 2005; Rieser et al., 2005).

Zollo collected seven dialogues with different human-wizard pairs to develop an evacuation plan. The overall WER was 30%. Of the 227 cases of incorrect ASR, wizard utterances indicated a failure to understand for only 35% of them. Wizards ignored words not salient in the domain and hypothesized words based on phonetic similarity. In (Skantze, 2003), both users and wizards knew there was no dialogue system; 44 direction-finding dialogues were collected with 16 subjects. Despite a WER of 43%, the wizard operators signaled misunderstanding only 5% of the time, in part because they often ignored ASR errors and continued the dialogue. For the 20% of non-understandings, operators continued a route description, asked a task-related question, or requested a clarification.

Williams and Young collected 144 dialogues simulating tourist requests for directions and other negotiations. WER was constrained to be high, medium, or low. Under medium WER, a task-related question in response to a non-understanding or misunderstanding led to full understanding more often than explicit repairs. Under high WER, however, the reverse was true. Misunderstandings significantly increased when wizards followed non-understandings or misunderstandings with a task-related question instead of a repair.

In (Rieser et al., 2005), wizards simulated a multimodal MP3 player application with access to a database of 150K music albums. Responses could be presented verbally or graphically. In the noisy transcription condition, wizards made clarification requests about twice as often as that found in similar human-human dialogue.

In a system like CheckItOut, user utterances that request database information must be understood. We seek an approach that would reduce the rate of misunderstandings observed for high WER in (Williams & Young, 2004) and the rate of clarification requests observed in (Rieser et al., 2005).

4 CheckItOut and Embedded Wizards

CheckItOut is modeled on library transactions at the Andrew Heiskell Braille and Talking Book Library, a branch of the New York Public Library and part of the National Library of Congress. Borrowing requests are handled by telephone. Books, mainly in a proprietary audio format, travel by mail. In a dialogue with CheckItOut, a user identifies herself, requests books, and is told which are available for immediate shipment or will go on reserve. The user can request a book by catalogue number, title, or author.

CheckItOut builds on the Olympus/RavenClaw framework (Bohus & Rudnicky, 2009) that has been the basis for about a dozen dialogue systems in different domains, including Let's Go Public! (Raux et al., 2005). Speech recognition relies on PocketSphinx. Phoenix, a robust context-free grammar (*CFG*) semantic parser, handles natural language understanding (Ward & Issar, 1994). The Apollo interaction manager (Raux & Eskenazi, 2007) detects utterance boundaries using information from speech recognition, semantic parsing, and Helios, an utterance-level confidence annotator (Bohus & Rudnicky, 2002). The dialogue manager is implemented in RavenClaw.

To design CheckItOut's dialogue manager, we recorded 175 calls (4.5 hours) from patrons to librarians. We identified 82 book request calls, transcribed them, aligned the utterances with the speech signal, and annotated the transcripts for dialogue acts. Because active patrons receive monthly newsletters listing new titles in the desired formats, patrons request specific items with advance knowledge of the author, title, or catalogue number. Most book title requests accurately reproduce the exact title, the title less an initial determiner ("the," "a"), or a subtitle.

We exploited the Galaxy message passing architecture of Olympus/RavenClaw to insert a wizard server into CheckItOut. The hub passes messages between the system and a wizard's graphical user interface *(GUI)*, allowing us to collect runtime information that can be included in models of wizards' actions.

For speech recognition, CheckItOut relies on PocketSphinx 0.5, a Hidden Markov Model-based recognizer. Speech recognition for this experiment, relied on the freely available Wall Street Journal "read speech" acoustic models. We did not adapt the models to our population or to spontaneous speech, thus insuring that wizards would receive relatively noisy recognition output.

We built trigram language models from the book titles using the CMU Statistical Language Modeling Toolkit. Pilot tests with one male and one female native speaker indicated that a language model based on 7500 titles would yield WER in the desired range. (Average WER for the book title requests in our experiment was 71%.) To model one aspect of the real world useful for an actual system, titles with below average circulation were eliminated. An offline pilot study had demonstrated that one-word titles were easy for wizards, so we eliminated those as well. A random sample of 7,500 was chosen from the remaining 19,708 titles to build the trigram language model.

We used Ratcliff/Obersherhelp (*R/O*) to measure the similarity of an ASR string to book titles in the database (Ratcliff & Metzener, 1988). R/O calculates the ratio r of the number of matching characters to the total length of both strings, but requires $O(r^2)$ time on average and $O(r^3)$ time in the worst case. We therefore computed an upper bound on the similarity of a title/ASR pair prior to full R/O to speed processing.

5 Experimental Design

In this experiment, a user and a wizard sat in separate rooms where they could not overhear one another. Each had a headset with microphone and a GUI. Audio input on the wizard's headset was disabled. When the user requested a title, the ASR hypothesis for the title appeared on the wizard's GUI. The wizard then selected the ASR hypothesis to execute a voice search against the database.

Given the ASR and the query return, the wizard's task was to guess which candidate in the query return, if any, matched the ASR hypothesis. Voice search accessed the full backend of 71,166 titles. The custom query designed for the experiment produced four types of return, in real time, based on R/O scores:

- *Singleton*: a single best candidate (R/O $\geq$ 0.85)
- *AmbiguousList*: two to five moderately good candidates (0.85 > R/O $\geq$ 0.55)
- *NoisyList*: six to ten poor but non-random candidates (0.55 > R/O $\geq$ 0.40)
- *Empty*: No candidate titles (max R/O < 0.40)

In pilot tests, 5%-10% of returns were empty versus none in the experiment. The distribution of other returns was: 46.7% Singleton, 50.5% AmbiguousList, and 2.8% NoisyList.

Seven undergraduate computer science majors at Hunter College participated. Two were non-native speakers of English (one Spanish, one Romanian). Each of the possible 21 pairs of students met for five trials. During each trial, one student served as wizard and the other as user for a *session* of 20 title cycles. They immediately reversed roles for a second session, as discussed further below. The experiment yielded 4172 title cycles rather than the full 4200, because users were permitted to end sessions early. All titles were selected from the 7500 used to construct the language model.

Each user received a printed list of 20 titles and a brief synopsis of each book. The acoustic quality of titles read individually from a list is unlikely to approximate that of a patron asking for a specific title. Therefore, immediately before each session, the user was asked to read a synopsis of each book, and to reorder the titles to reflect some logical grouping, such as genre or topic. Users requested titles in this new order that they had created.

Participants were encouraged to maximize a session score, with a reward for the experiment winner. Scoring was designed to foster cooperative strategies. The wizard scored +1 for a correctly identified title, +0.5 for a thoughtful question, and -1 for an incorrect title. The user scored +0.5 for a successfully recognized title. User and wizard traded roles for the second session, to discourage participants from sabotaging the others' scores.

The wizard's GUI presented a real-time live feed of ASR hypotheses, weighted by grayscale to reflect acoustic confidence. Words in each candidate title that matched a word in the ASR appeared darker: dark black for Singleton or AmbiguousList, and medium black for NoisyList. All other words were in grayscale in proportion to the degree of character overlap. The wizard queried the database with a recognition hypothesis for one utterance at a time, but could concatenate successive utterances, possibly with some limited editing.

After a query, the wizard's GUI displayed candidate matches in descending order of R/O score. The wizard had four options: make a *firm choice* of a candidate, make a *tentative choice*, ask a *question*, or *give up* to end the title cycle. Questions were recorded. The wizard's GUI showed the success or failure of each title cycle before the next one began. The user's GUI posted the 20 titles to be read during the session. On the GUI, the user rated the wizard's title choices as correct or incorrect. Titles were highlighted green if the user judged a wizard's offered title correct, red if incorrect, yellow if in progress, and not highlighted if still pending. The user also rated the wizard's questions. Average elapsed time for each 20-title session was 15.5 minutes.

A questionnaire similar to the type used in PARADISE evaluations (Walker et al., 1998) was administered to wizards and users for each pair of sessions. On a 5-point Likert scale, the average response to the question "I found the system easy to use this time" was 4 (sd=0; 4=Agree), indicating that participants were comfortable with the task. All other questions received an average score of Neutral (3) or Disagree (2). For example, participants were neutral (3) regarding confidence in guessing the correct title, and disagreed (2) that they became more confident as time went on.

6 Learning Method and Goals

To model wizard actions, we assembled 60 features that would be available at run time. Part of our task was to detect their relative independence, meaningfulness, and predictive ability. Features described the wizard's GUI, the current title session, similarity between ASR and candidates, ASR relevance to the database, and recognition and confidence measures. Because the number of voice search returns varied from one title to the next, features pertaining to candidates were averaged.

We used three machine-learning techniques to predict wizards' actions: decision trees, linear regression, and logistic regression. All models were produced with the Weka data mining package, using 10-fold cross-validation (Witten & Frank, 2005). A decision tree is a predictive model that maps feature values to a target value. One applies a decision tree by tracing a path from the *root* (the top node) to a leaf, which provides the target value. Here the leaves are the wizard actions: firm choice, tentative choice, question, or give up. The algorithm used is a version of C4.5 (Quinlan, 1993), where gain ratio is the splitting criterion.

To confirm the learnability and quality of the decision tree models, we also trained logistic regression and linear regression models on the same data, normalized in [0, 1]. The logistic regression model predicts the probability of wizards' actions by fitting the data to a logistic curve. It generalizes the linear model to the prediction of categorical data; here, categories correspond to wizards' actions. The linear regression models represent wizards' actions numerically, in decreasing value: firm choice, tentative choice, question, give up.

Although analysis of individual wizards has not been systematic in other work, we consider the variation in human performance significant. Because we seek excellent, not average, teachers for CheckItOut, our focus is on understanding good wizardry. Therefore, we learned two kinds of models with each of the three methods: the *overall model* using data from all of our wizards, and individual *wizard models*.

Preliminary cross-correlation confirmed that many of the 60 features were heavily interdependent. Through an initial manual curation phase, we isolated groups of features with $R^2 > 0.5$. When these groups referenced semantically similar features, we selected a single representative from the group and retained only that one. For example, the features that described similarity between hypotheses and candidates were highly correlated, so we chose the most comprehensive one: the number of exact word matches. We also grouped together

Table 1. Raw session score, accuracy, proportion of offered titles that were listed first in the query return, and frequency of correct non-offers for seven participants.

Participant	Cycles	Session Score	Accuracy	Offered Return 1	Correct Non-Offers
W4	600	0.7585	0.8550	0.70	0.64
W5	600	0.7584	0.8133	0.76	0.43
W7	599	0.6971	0.7346	0.76	0.14
W1	593	0.6936	0.7319	0.79	0.16
W2	599	0.6703	0.7212	0.74	0.10
W3	581	0.6648	0.6954	0.81	0.20
W6	600	0.6103	0.6950	0.86	0.03

and represented by a single feature: three features that described the gaps between exact word matches, three that described the data presented to the wizard, nine that described various system confidence scores, and three that described the user's speaking rate. This left 28 features.

Next we ran CfsSubsetEval, a supervised attribute selection algorithm for each model (Witten & Frank, 2005). This greedy, hill-climbing algorithm with backtracking evaluates a subset of attributes by the predictive ability of each feature and the degree of redundancy among them. This process further reduced the 28 features to 8-12 features per model. Finally, to reduce overfitting for decision trees, we used pruning and subtree rising. For linear regression we used the M5 method, repeatedly removing the attribute with the smallest standardized coefficient until there was no further improvement in the error estimate given by the Akaike information criterion.

7 Results

Table 1 shows the number of title cycles per wizard, the raw session score according to the formula given to the wizards, and accuracy. *Accuracy* is the proportion of title cycles where the wizard found the correct title, or correctly guessed that the correct title was not present (asked a question or gave up). Note that score and accuracy are highly correlated (R=0.91, p=0.0041), indicating that the instructions to participants elicited behavior consistent with what we wanted to measure.

Wizards clearly differed in performance, largely due to their response when the candidate list did not include the correct title. Analysis of variance with wizard as predictor and accuracy as the dependent variable is highly significant (p=0.0006); significance is somewhat greater (p=0.0001) where session score is the dependent variable. Table 2 shows the distribution of correct actions: to offer a candidate at a given position in the query return (Returns 1 through 9), or to ask a question or give up. As reflected in Table 2, a baseline accuracy of about 65% could be achieved by offering the first return. The fifth column of Table 1 shows how often wizards did that (Offered Return 1), and clearly illustrates that those who did so most often (W3 and W6) had accuracy results closest to the baseline. The wizard who did so least often (W4) had the highest accuracy, primarily because she more often correctly offered no title, as shown in the last column of Table 1. We conclude that a spoken dialogue system would do well to emulate W4.

Overall, our results in modeling wizards' actions were uniform across the three learning methods, gauged by accuracy and F measure. For the combined wizard data, logistic regression had an accuracy of 75.2%, and F measures of 0.83 for firm choices and 0.72 for tentative choices; the decision tree accuracy was 82.2%, and the F measures for firm versus tentative choices were respectively 0.82 and 0.71. The decision tree had a root mean squared error of 0.306, linear regression 0.483. Table 3 shows the accuracy and F measures on firm choices for the decision trees by individual wizard, along with the numbers of attributes and nodes per

Table 2. Distribution of correct actions

Correct Action	N	%
Return 1	2722	65.2445
Return 2	126	3.0201
Return 3	56	1.3423
Return 4	46	1.1026
Return 5	26	0.6232
Return 7	7	0.1678
Return 8	1	0.0002
Return 9	2	0.0005
Question or Giveup	1186	28.4276
Total	4172	1.0000

Table 3. Learning results for wizards

Tree	Rank	Nodes	Attributes	Accuracy	F firm
W4	1	55	12	75.67	0.85
W5	2	21	10	76.17	0.85
W1	3	7	8	80.44	0.87
W7	4	45	11	73.62	0.83
W3	5	33	10	77.42	0.84
W2	6	35	10	78.49	0.85
W6	7	23	10	85.19	0.80

tree. Although relatively few attributes appeared in any one tree, most attributes appeared in multiple nodes. W1 was the exception, with a very small pruned tree of 7 nodes.

Accuracy of the decision trees does not correlate with wizard rank. In general, the decision trees could consistently predict a confident choice (0.80 ≤ F ≤ 0.87), but were less consistent on a tentative choice (0.60 ≤ F ≤ 0.89), and could predict a question only for W4, the wizard with the highest accuracy and greatest success at detecting when the correct title was not in the candidates.

What wizards saw on the GUI, their recent success, and recognizer confidence scores were key attributes in the decision trees. The five features that appeared most often in the root and top-level nodes of all tree models reported in Table 3 were:

- *DisplayType* of the return (Singleton, Ambiguous List, NoisyList)
- *RecentSuccess,* how often the wizard chose the correct title within the last three title cycles
- *ContiguousWordMatch,* the maximum number of contiguous exact word matches between a candidate and the ASR hypothesis (averaged across candidates)
- *NumberOfCandidates*, how many titles were returned by the voice search
- *Confidence,* the Helios confidence score

DisplayType, NumberOfCandidates and *ContiguousWordMatch* pertain to what the wizard could see on her GUI. (Recall that *DisplayType* is distinguished by font darkness, as well as by number of candidates.) The impact of *RecentSuccess* might result not just from the wizard's confidence in her current strategy, but also from consistency in the user's speech characteristics. The Helios confidence annotation uses a learned model based on features from the recognizer, the parser, and the dialogue state. Here confidence primarily reflects recognition confidence; due to the simplicity of our grammar, parse results only indicate whether there is a parse. In addition to these five features, every tree relied on at least one measure of similarity between the hypothesis and the candidates.

W4 achieved superior accuracy: she knew when to offer a title and when not to. In the learned tree for W4, if the *DisplayType* was *NoisyList*, W4 asked a question; if *DisplayType* was *AmbiguousList,* the features used to predict W4's action included the five listed above, along with the acoustic model score, word length of the ASR, number of times the wizard had asked the user to repeat, and the maximum size of the gap between words in the candidates that matched the ASR hypothesis.

To focus on W4's questioning behavior, we trained an additional decision tree to learn how W4 chose between two actions: offering a title versus asking a question. This 37-node, 8-attribute tree was based on 600 data points, with F=0.91 for making an offer and F=0.68 for asking a question. The tree is distinctive in that it splits at the root on the number of frames in the ASR. If the ASR is short (as measured both by the number of recognition frames and the words), W4 asks a question when *DisplayType* = *AmbiguousList* or *NoisyList,* either *RecentSuccess* ≤ 1 or *ContiguousWordMatch* = 0, and the acoustic model score is low. Note that shorter titles are more confusable. If the ASR is long, W4 asks a question when *ContiguousWordMatch* ≤ 1, *RecentSuccess* ≤ 2, and either *CandidateDisplay* = *NoisyList*, or *Confidence* is low, and there is a choice of titles.

8 Discussion

Our experiment addressed whether voice search can compensate for incorrect ASR hypotheses and permit identification of a user's desired book, given a request by title. The results show that with high WER, a baseline dialogue strategy that always offers the highest-ranked database return can nevertheless achieve moderate accuracy. This is true even with the relatively simplistic measure of similarity between the ASR hypothesis and candidate titles used here. As a result, we have integrated voice search into CheckItOut, along with a linguistically motivated grammar for book titles. Our current Phoenix grammar relies on CFG rules automatically generated from dependency parses of the book titles, using the MICA parser

(Bangalore et al., 2009). As described in (Gordon & Passonneau, 2010), a book title parse can contain multiple title slots that consume discontinuous sequences of words from the ASR hypothesis, thus accommodating noisy ASR. For the voice search phase, we now concatenate the words consumed by a sequence of title slots. We are also experimenting with a statistical machine learning approach that will replace or complement the semantic parsing.

Computers clearly do some tasks faster and more accurately than people, including database search. To benefit from such strengths, a dialogue system should also accommodate human preferences in dialogue strategy. Previous work has shown that user satisfaction depends in part on task success, but also on minimizing behaviors that can increase task success but require the user to correct the system (Litman et al., 2006).

The decision tree that models W4 has lower accuracy than other models' (see Table 3), in part because her decisions had finer granularity. A spoken dialogue system could potentially do as well as or better than the best human at detecting when the title is not present, given the proper training data. To support this, a dataset could be created that was biased toward a larger proportion of cases where not offering a candidate is the correct action.

9 Conclusion and Current Work

This paper presents a novel methodology that embeds wizards in a spoken dialogue system, and collects data for a single turn exchange. Our results illustrate the merits of ranking wizards, and learning from the best. Our wizards were uniformly good at choosing the correct title when it was present, but most were overly eager to identify a title when it was not among the candidates. In this respect, the best wizard (W4) achieved the highest accuracy because she demonstrated a much greater ability to know when *not* to offer a title. We have shown that it is feasible to replicate this ability in a model learned from features that include the presentation of the search results (length of the candidate list, amount of word overlap of candidates with the ASR hypothesis), recent success at selecting the correct candidate, and measures pertaining to recognition results (confidence, acoustic model score, speaker rate). If replicated in a spoken dialogue system, such a model could support integration of voice search in a way that avoids misunderstandings. We conclude that learning from embedded wizards can exploit a wider range of relevant features, that dialogue managers can profit from access to more fine-grained representations of user utterances, and that machine learners should be selective about which people to model.

That wizard actions can be modeled using system features bodes well for future work. Our next experiment will collect full dialogues with embedded wizards whose actions will again be restricted through an interface. This time, NLU will integrate voice search with the linguistically motivated CFG rules for book titles described earlier, and a larger language model and grammar for database entities. We will select wizards who perform well during pilot tests. Again, the goal will be to model the most successful wizards, based upon data from recognition results, NLU, and voice search results.

Acknowledgements

This research was supported by the National Science Foundation under IIS-0745369, IIS-084966, and IIS-0744904. We thank the anonymous reviewers, the Heiskell Library, our CMU collaborators, our statistical wizard Liana Epstein, and our enthusiastic undergraduate research assistants.

References

Bangalore, Srinivas; Bouillier, Pierre; Nasr, Alexis; Rambow, Owen; Sagot, Benoit (2009). *MICA: a probabilistic dependency parser based on tree insertion grammars. Application Note.* Human Language Technology and North American Chapter of the Association for Computational Linguistics, pp. 185-188.

Bohus, D.; Rudnicky, A.I. (2009). The RavenClaw dialog management framework: Architecture and systems. *Computer Speech and Language, 23*(3), 332-361.

Bohus, Daniel; Rudnicky, Alex (2002). *Integrating multiple knowledge sources for utterance-level confidence annotation in the CMU Communicator spoken dialog system* (Technical Report No. CS-190): Carnegie Mellon University.

Georgila, Kallirroi; Sgarbas, Kyrakos; Tsopanoglou, Anastasios; Fakotakis, Nikos; Kokkinakis, George (2003). A speech-based human-computer interaction system for automating directory assistance services. *International Journal of Speech Technology, Special Issue on Speech and Human-Computer Interaction, 6*(2), 145-59.

Gordon, Joshua, B.; Passonneau, Rebecca J. (2010). *An evaluation framework for natural language understanding in spoken dialogue systems*. Seventh International Conference on Language Resources and Evaluation (LREC).

Johnston, Michael; Bangalore, Srinivas; Vasireddy, Gunaranjan; Stent, Amanda; Ehlen, Patrick; Walker, Marilyn A., et al. (2002). *MATCH--An architecture for multimodal dialogue systems*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 376-83.

Komatani, Kazunori; Kanda, Naoyuki; Ogata, Tetsuya; Okuno, Hiroshi G. (2005). *Contextual constraints based on dialogue models in database search task for spoken dialogue systems*. The Ninth European Conference on Speech Communication and Technology (Eurospeech), pp. 877-880.

Kruijff-Korbayová, Ivana; Blaylock, Nate; Gerstenberger, Ciprian; Rieser, Verena; Becker, Tilman; Kaisser, Michael, et al. (2005). *An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system*. 10th European Workshop on Natural Language Generation (ENLG), pp. 191-196.

Levin, Esther; Narayanan, Shrikanth; Pieraccini, Roberto; Biatov, Konstantin; Bocchieri, E.; De Fabbrizio, Giuseppe, et al. (2000). *The AT&T-DARPA Communicator Mixed-Initiative Spoken Dialog System*. Sixth International Conference on Spoken Dialogue Processing (ICLSP), pp. 122-125.

Litman, Diane; Hirschberg, Julia; Swerts, Marc (2006). Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics, 32*(3), 417-438.

Litman, Diane; Pan, Shimei (1999). *Empirically evaluating an adaptable spoken dialogue system*. 7th International Conference on User Modeling (UM), pp. 55-46.

Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Ratcliff, John W.; Metzener, David (1988). Pattern Matching: The Gestalt Approach. *Dr. Dobb's Journal*, 46

Raux, Antoine; Bohus, Dan; Langner, Brian; Black, Alan W.; Eskenazi, Maxine (2006). *Doing research on a deployed spoken dialogue system: one year of Let's Go! experience*. Ninth International Conference on Spoken Language Processing (Interspeech/ICSLP).

Raux, Antoine; Eskenazi, Maxine (2007). *A Multi-layer architecture for semi-synchronous event-driven dialogue management.*IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2007), Kyoto, Japan.

Raux, Antoine; Langner, Brian; Black, Alan W.; Eskenazi, Maxine (2005). *Let's Go Public! Taking a spoken dialog system to the real world.*Interspeech 2005 (Eurospeech), Lisbon, Portugal.

Rieser, Verena; Kruijff-Korbayová, Ivana; Lemon, Oliver (2005). *A corpus collection and annotation framework for learning multimodal clarification strategies*. Sixth SIGdial Workshop on Discourse and Dialogue, pp. 97-106.

Sacks, Harvey; Schegloff, Emanuel A.; Jefferson, Gail (1974). A simplest systematics for the organization of turn-taking for conversation. *Language, 50*(4), 696-735.

Skantze, Gabriel (2003). *Exploring human error handling strategies: Implications for Spoken Dialogue Systems*. Proceedings of ISCA Tutorial and Research Workshp on Error Handling in Spoken Dialogue Systems, pp. 71-76.

Stoyanchev, Svetlana; Stent, Amanda (2009). *Predicting concept types in user corrections in dialog*. Proceedings of the EACL Workshop SRSL 2009, the Second Workshop on Semantic Representation of Spoken Language, pp. 42-49.

Turunen, Markku; Hakulinen, Jaakko; Kainulainen, Anssi (2006). *Evaluation of a spoken dialogue system with usability tests and long-term pilot studies*. Ninth International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP).

Walker, M A.; Litman, D, J.; Kamm, C. A.; Abella, A. (1998). Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies. *Computer Speech and Language, 12*, 317-348.

Wang, Ye-Yi; Yu, Dong; Ju, Yun-Cheng; Acero, Alex (2008). An introduction to voice search. *IEEE Signal Process. Magazine, 25*(3).

Ward, Wayne; Issar, Sunil (1994). *Recent improvements in the CMU spoken language understanding system.*ARPA Human Language Technology Workshop, Plainsboro, NJ.

Williams, Jason D.; Young, Steve (2004). *Characterising Task-oriented Dialog using a Simulated ASR Channel*. Eight International Conference on Spoken Language Processing (ICSLP/Interspeech), pp. 185-188.

Witten, Ian H.; Frank, Eibe (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Zollo, Teresa (1999). *A study of human dialogue strategies in the presence of speech recognition errors*. Proceedings of AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems, pp. 132-139.

Zue, Victor; Seneff, Stephanie; Glass, James; Polifroni, Joseph; Pao, Christine; Hazen, Timothy J., et al. (2000). A Telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing, 8*, 85-96.