

Concept Type Prediction and Responsive Adaptation in a Dialogue System

Svetlana Stoyanchev

SSTOYANCHEV@CS.COLUMBIA.EDU

Columbia University, Computer Science Department

450 Computer Science Building, 1214 Amsterdam Avenue, New York, NY 10027-7003, USA

Amanda J. Stent

STENT@RESEARCH.ATT.COM

AT&T Labs – Research

180 Park Avenue, Florham Park, NJ 07932, USA

Editor: Gregory Aist

Abstract

Responsive adaptation in spoken dialogue systems involves a change in dialogue system behavior in response to a user or a dialogue situation. In this paper we address responsive adaptation in the automatic speech recognition module of a spoken dialogue system. We hypothesize that information about the content of a user utterance may help improve speech recognition. We use a two-step process to test this hypothesis: first, we automatically predict the task-relevant concept types likely to be present in a user utterance using features from the dialogue context and from the output of first-pass recognition of the utterance; and then, we adapt the speech recognizer's language model to the predicted content of the user's utterance and run a second pass of speech recognition. We show that: (1) it is possible to achieve high accuracy in determining presence or absence of particular concept types in a post-confirmation utterance; and (2) 2-pass speech recognition with concept type classification and language model adaptation can lead to improved speech recognition performance for post-confirmation utterances.

Keywords: Speech recognition, Dialog structure, Error handling

1. Introduction

There are many possible sources of error in dialogue system processing, but the source that is most immediately obvious to the user is speech recognition errors. Furthermore, despite years of research on robust handling of errorful input, most dialogue systems still cannot adapt their behavior when the user responds in an unexpected way to a speech recognition error. Einstein said that the definition of insanity is doing the same thing and expecting a different result. In the presence of a speech recognition error, users frequently vary their input (changing the volume, pitch, speaking rate, words, syntax, or concepts) (Litman et al. 2006, Oviatt et al. 1998, Shin et al. 2002). However, the dialogue system typically does not change its expectations about the form of a response, leading to cascading errors, low task completion rates and poor levels of user satisfaction.

In this paper we propose a method that permits a dialogue system to responsively adapt its speech recognition behavior in the face of unexpected user input by using the dialogue context and information from the current user utterance. The proposed method uses two-pass speech recognition, with a concept type predictor that predicts the presence of task-relevant concepts in the user's utterance. The prediction happens between the two speech recognition passes. In the first pass the

speech recognizer uses a generic language model; then, the first-pass speech recognition results and other features are fed into the concept type predictor; and finally, the predicted concept types are used to select an adapted language model to be used in the second-pass speech recognition.

We evaluate this method at one crucial point in a dialogue: *post-confirmation utterances*, or user utterances made in response to system confirmation prompts (Section 4.2). Confirmation prompts are yes/no questions a system produces to solicit user confirmation that it has correctly recognized an earlier input. An expected behaviour in response to a confirmation prompt is a *yes* or *no* answer. However, our observations show that users often exhibit unexpected behaviour in post-confirmation utterances, providing extra information and specifying new task-relevant concepts, causing frequent failures in speech recognition. It is particularly important for systems to recognize post-confirmation prompts correctly because recognition errors at these dialogue locations lead to cascading error subdialogues, frustrating the user and negatively impacting task success.

We hypothesize that information about the content of the current user utterance as well as the dialogue history may lead to improved speech recognition. We use a two-step process to test this hypothesis. First, we test concept type prediction (Section 6), and second, we test speech recognition with an adapted language model (Section 7). In the concept type prediction experiment, we automatically predict the expected content of post-confirmation user utterances using features from the dialogue context and from the output of first-pass speech recognition. We show that it is possible to achieve high accuracy in determining the presence or absence of particular concept types in post-confirmation utterances. In the speech recognition experiment we adapt the speech recognizer’s language model to the predicted content of the user’s utterance and evaluate the accuracy of second-pass speech recognition. We show that two-pass speech recognition with concept type classification and language model adaptation can lead to improved speech recognition performance.

The rest of this paper is structured as follows: in Section 2, we define the notion of responsive adaptation and motivate this research. In Section 3, we outline related work. In Section 4, we describe the system and data that we used and in Section 5, our experimental method. In Sections 6 and 7 we present our experimental results for the concept type prediction and speech recognition experiments. Finally, in Sections 8 and 9 we conclude and present ideas for future work.

2. Motivation

Adaptation is a natural and effective behavior, widely used by humans in spoken dialogue with each other and with computers (Brennan and Clark 1996, Kraljic et al. 2008, Garrod and Anderson 1987, Branigan et al. 2004, Dubey et al. 2006b). However, most dialogue systems do not adapt their behavior. Furthermore, when a dialogue system does adapt the adaptation is typically *egocentric* or *directive*. *Egocentric adaptation* takes place when a dialogue system changes its behavior in response to a change in its own internal state (e.g. when a dialogue system changes its responses because of its internal state of misunderstanding (Hockey et al. 2003)). *Directive adaptation* takes place when a dialogue system attempts to cause a change in user behavior (e.g. by requesting input in a different modality or different manner (Filisko and Seneff 2005), or by modifying output to cause changes in the user’s input style (Kruijff-Korbayova and Kukina 2008)). By contrast, *responsive adaptation* in spoken dialogue systems involves a change in dialogue system behavior in direct response to a user input or a dialogue situation. The change can be manifested in any of the components of a dialogue system: natural language understanding (NLU), natural language generation (NLG), dialogue management (DM), or speech recognition (ASR). For example, a system

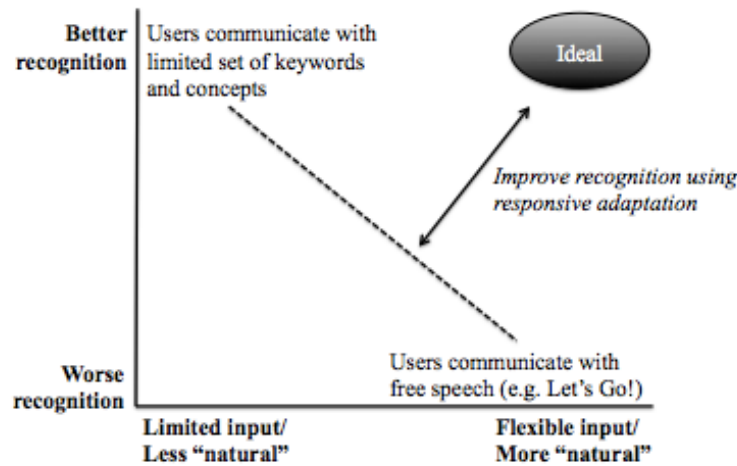


Figure 1: Dialogue systems recognition and interaction

may modify its parsing lexicon and grammar based on the words and constructs used by the user (Dubey et al. 2006a), or may modify its generation strategy based on a model of the user’s expertise or certainty (Fukubayashi et al. 2006, Forbes-Riley and Litman 2009). Because speech recognition is the component closest to the user, the impact of responsive adaptation is most easily explored in the context of adaptation in the speech recognizer.

The motivation for our two-stage adaptive recognition approach is drawn from human language processing, in which the context of an utterance helps conversation partners disambiguate speech. For example, “*Take this train*”, “*Take the strain*” and even “*Take this drain*” can in normal conversational speech only be distinguished from each other by context¹. A dialogue system faces a similar challenge; for example, the place name “*Admore*” may be misrecognized as the more frequently occurring common noun “*morning*” in the *Let’s Go!* dialogue system.

Speech recognizers use two models: an acoustic model and a language model. The acoustic model, which maps acoustic frequency features to lexical units, is generated from speech data with aligned transcriptions. The language model scores sequences of lexical units using a model of the likelihood of their component n-grams (word sequences of length n). Language models can be statistical (generated from a text corpus) or grammar-based (generated from a manually constructed context free grammar). This means that the goodness of fit between user utterances and the dataset or grammar used to generate the language model affects the performance of the speech recognizer.

Word error rates for commercial state-of-the-art open-domain speaker-independent speech recognition technology are around 25%-30% (Riccardi and Hakkani-Tür 2003). Noisy conditions, speaker accent, and out-of-vocabulary speech are among many factors that may increase the frequency of recognition errors. The performance of speech recognition is also partly dependent on the type of input the system is designed to recognize (see Figure 1). *Limited input* dialogue systems require the user to respond to each system prompt using only the concepts and keywords currently requested by the system. By contrast, *flexible input* dialogue systems allow the user to respond to system prompts with longer phrases and sentences and specify information other than that currently re-

1. We thank Alistair Conkie for this example.

quested. Speech recognition (ASR) accuracy in *limited input* systems is better than in *flexible input* systems (Danieli and Gerbino 1995, Smith and Gordon 1997). However, task completion rates and times can be better in *flexible input* systems (Chu-Carroll and Nickerson 2000, Smith and Gordon 1997). Researchers have shown that user training improves performance of *limited input* systems, while prompt design improves performance of *flexible input* systems. For example, Tomko and Rosenfeld showed that trained users communicating with a *limited input* dialogue system achieve better speech recognition than users communicating with a corresponding *flexible input* dialogue system (Tomko and Rosenfeld 2006). Sheeder and Balogh showed that in *flexible input* dialogue systems prompts can be formulated to maximize speech recognition accuracy and reduce the number of speech recognition timeouts (Sheeder and Balogh 2003).

It is now common practice to adapt speech recognizers to the type, context or style of input speech (Bellegarda 2004). Language model adaptation has been used to improve automatic speech recognition performance in automated meeting transcription (Tur and Stolcke 2007), speech-driven question answering (Stoyanchev et al. 2008), broadcast news recognition (Gildea and Hofmann 1999), and spoken dialogue systems (Tur et al. 2005). Language models in dialogue systems can be adapted to the dialogue state (Riccardi and Gorin 2000, Esteve et al. 2001), the topic (Iyer and Ostendorf 1999, Gildea and Hofmann 1999), or the speaker (Tur 2007). However, typically the language model is adapted based on dialogue system behavior (e.g. the topic of the system’s prompt) rather than on user behavior. Language model adaptation in our work is based on user behavior; in particular, the content of the user’s current utterance and the dialogue context.

3. Related Work on Adaptation in Speech Recognition

Language model adaptation is a technique for improving speech recognition performance. It involves adjusting probabilities in the language model or selecting the data for building the language model. The goal of this adaptation is to make the model better fit the language in input utterances, leading to improved speech recognition.

Riccardi and Gorin (2000) describe an approach to language model adaptation in which the language model is conditioned on the current state of the dialogue system, leading to reductions in word error rate. It has now become standard practice to use dialogue state-specific language models (Bechet et al. 2004), and the system we used for our experiments follows this approach.

Iyer and Ostendorf (1999) describe an approach to language model adaptation based on topic rather than on dialogue state. By using a weighted combination of topic-specific language models, they obtained a 4.5% reduction in word error rate on the Wall Street Journal text corpus, but only a 1.2% relative reduction in word error rate on the Switchboard spoken dialogue corpus. In other work on topic-based language model adaptation, Martins et al. (2010) dynamically adapt a language model for broadcast news recognition over time by using documents retrieved from the Web.

Co-constraining speech recognition and natural language understanding (NLU) has been shown to benefit both processes. Young (1994) uses output from the NLU along with acoustic model probabilities to detect misrecognized words on a second pass through the recognizer. Bigi et al. (2004) describe another two-pass approach to speech recognition. The authors use terms from first-pass speech recognition to retrieve matching documents, and then interpolate a generic language model with a model built on the retrieved documents.

Language model adaptation can be most easily done for statistical language models. However, many dialogue systems use grammar-based language models, which can be created in the absence of

training data. Grammar-based and statistical language modeling can be combined to improve speech recognition performance. In some research, a probabilistic grammar is used directly (e.g. Jurafsky et al. (1995), Knight et al. (2001)). By contrast, Gorrell et al. (2002) and Hockey et al. (2003) use a combination of grammar-based and statistical speech recognition in a two-pass approach. First, the user’s utterance is passed through a grammar-based language model (LM). Using a threshold on confidence level, the system either accepts the utterance or passes it to a statistical LM.

In our experiments we also perform language model adaptation in a two-stage speech recognition approach. Instead of performing language model interpolation based on document retrieval, we perform language model selection based on concept type prediction. Our concept type predictor uses features similar to those used by Gabsdil and Lemon (2004) and Litman et al. (2006). We specifically address speech recognition of user utterances following system confirmation prompts. In the cases when a user attempts to correct the system in a post-confirmation utterance, we observe a significant increase in word error rate for the user’s utterance, which may lead to a cascading error sequence of misunderstandings. By adapting the system’s language model to the predicted task-related concepts in the user’s utterance, we achieve improved speech recognition, limiting the likelihood of cascading errors. This method operates independently of the type of language model used or the amount of training data in the language model.

4. System

In our experiments we used the *Let’s Go!* dialogue system (Raux et al. 2005). In this section we briefly describe *Let’s Go!*, and then discuss the task-related concepts and confirmation types present in the system.

4.1 Let’s Go System Description

Let’s Go! is a telephone-based dialogue system maintained and deployed at Carnegie Mellon University. It provides information about bus routes in Pittsburgh. The system is reachable through the local Port Authority number outside of business hours (human operators answer the phone lines during business hours). It consequently receives calls from a diverse population of real users.

Let’s Go! was developed using the Olympus distributed dialogue framework and has the architecture shown in Figure 2. System components run as separate applications communicating through a central hub. Speech recognition is done by the Pocket Sphinx speech recognizer (Huggins-Daines et al. 2006). Speech recognition output is parsed by Phoenix, a robust parser which allows the system to skip unknown words and perform partial parsing (Ward and Issar 1994a). The dialogue manager was developed using RavenClaw (Bohus and Rudnicky 2003); in RavenClaw, dialogue structure is defined as a graph whose nodes are minimal dialogue components (typically individual exchanges), and whose edges indicate dialogue flow. The Rosetta template-based generator is used for response generation. The speech synthesis component is the open source Free TTS system. Other research Olympus-based dialogue systems developed at CMU include the Room Line, LARRI (Bohus and Rudnicky 2002), and SBU Calendar (Stenchikova et al. 2007) systems². The *Let’s Go!* domain and data were used in the first and second spoken dialogue system challenges (Black et al. 2010).

2. The SBU Calendar system was the first dialogue system based on the Ravenclaw/Olympus framework to be developed outside of CMU.

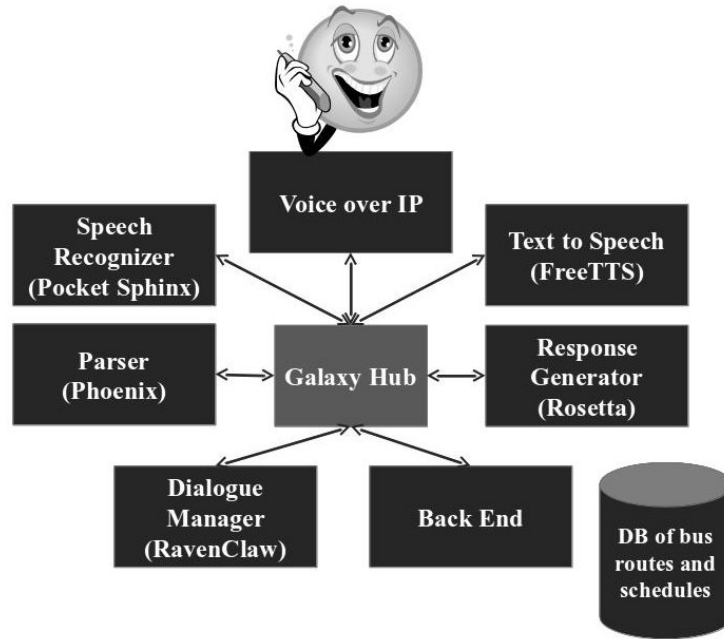


Figure 2: *Let's Go!* system architecture

| | |
|---------|--|
| SYSTEM: | Welcome to the CMU Lets Go! Bus Information System. How may I help you? |
| USER: | I'd like to go to Forbes and Murray. |
| SYSTEM: | To Forbes and Murray. Where are you leaving from? |
| USER: | Carnegie Mellon. |
| SYSTEM: | From Carnegie Mellon. There are five buses running between Carnegie Mellon and Forbes and Murray: the 501, the 59U, the 61A, the 61B, and the 61C. What else can I do for you? |
| USER: | When is the next one? |
| SYSTEM: | The next 61C going to Forbes and Murray will leave Carnegie Mellon at 5:13 PM. |

Table 1: Sample dialogue with *Let's Go!*

In the two datasets we analyzed from 2005 and 2006, *Let's Go!* received on average 40 calls per day. Average call length was 12.9 turns, but there was a large standard deviation in call length. A 2005 call analysis showed a raw speech recognition word error rate of 68% (Raux et al. 2005). The task success rate was estimated at 43%. Table 1 shows a sample dialogue with the system.

To accommodate the diverse user population and noisy speaking conditions, *Let's Go!* is designed as a flexible-input, linear system-initiative dialogue following an initial open prompt (*How may I help you?*). In order to provide the user with route information, *Let's Go!* elicits values for four task-related concepts: a departure location, a destination, a departure time, and optionally a bus route number. Each concept value provided by the user is explicitly confirmed by the system.

| Concept type | Example user utterance |
|--------------|-------------------------------------|
| place | I need to go from Oakland: p |
| time | Leaving at four p. m.: t |
| bus | I need 28X: b |

Table 2: Examples of concept-containing user utterances to the *Let's Go!* system. Concept annotations: **:p** indicates place, **:t** indicates time, and **:b** indicates bus.

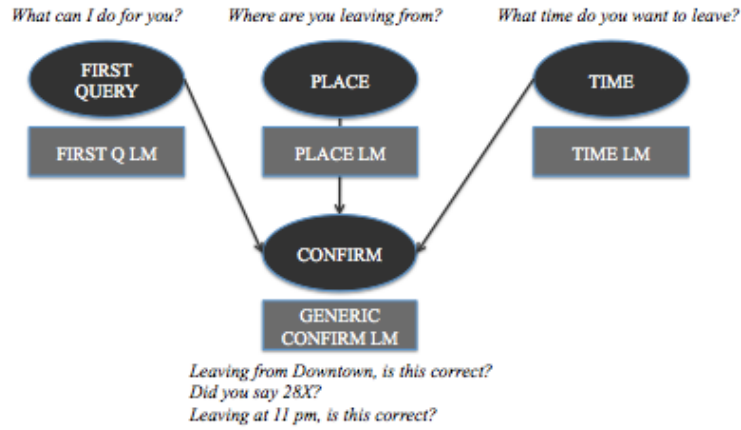


Figure 3: Dialogue states and language models used in *Let's Go!*

Let's Go! adapts its language model turn-by-turn according to the system's dialogue state. It has four dialogue states corresponding to the information it elicits: *first-query*, *place*, *time*, and *confirm* (see Figure 3)³. The state-specific language models are trained on user utterances from the corresponding dialogue states from previous dialogues with the system. For example, the **PLACE LM** is built from responses to the *Where are you leaving from?* prompt. The **PLACE LM** is more likely to correctly recognize typical user responses with location-relevant vocabulary such as *leaving*, *from*, *going*, *to* and place names⁴. It is less likely to correctly recognize responses containing other task-relevant concepts (e.g. *at four*).

4.2 Concepts and Confirmations

Let's Go! is a flexible input system. It allows users to specify any combination of concepts in each state. For example, in response to a *first query* prompt the user can specify all of the information about the desired route (e.g. *Going from Downtown to Oakland at four p.m.*), or only part of the information (e.g. *Leaving from Downtown*). In response to the *place* prompt, *Where are you leaving from?*, users are likely to specify a place concept value; however, they can also take task initiative and specify values for other concepts. Users can even specify a concept for which there is no state. Although there is no state corresponding to a *bus route* request (as the system does not ask the user

3. There is also *next-query* which is similar to *first-query* and is omitted from the diagram.

4. Language models used by *Let's Go!* are hierarchical. Concept values such as place names are stored in a dictionary. If the training data contains an utterance with a place concept, the relevant concept values are listed in a subsidiary language model inserted at the location of the place concept.

| System's confirmation question | User response | Response type |
|--|--------------------------------|-----------------------------------|
| Going to WOOD STREET. Did I get that right? | yes | <i>Positive confirmation</i> |
| Leaving from DOWNTOWN. Did I get that right? | no, Oakland | <i>Rejection & correction</i> |
| Leaving from Waterfront, is this correct? | yes and go to Oakland | <i>Topic change</i> |
| Leaving from ROBINSON. Is this correct? | from Polish Hill | <i>Correction</i> |
| Going to REGENT SQUARE. Is this correct? | no, Braddock avenue | <i>Rejection & correction</i> |
| The 61A. Did I get that right? | wondering when the next bus is | <i>Topic change</i> |

Table 3: Example answers to system confirmation prompts

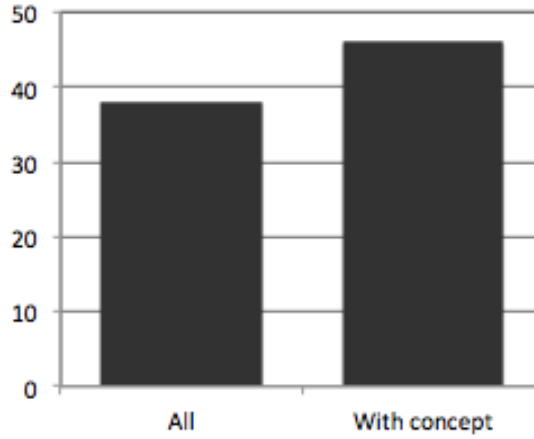


Figure 4: Word error rate on post-confirmation user utterances

for a bus route explicitly), a bus route can be specified in responses to other prompts. For example, after the *How may I help you?* prompt the user may respond *I want to take a 28X*.⁵

Users are particularly likely to specify non-requested concept values in responses to system confirmation prompts. *Let's Go!*, like most dialogue systems, explicitly confirms user-provided task-related concepts. The user's response to a confirmation prompt such as *Leaving from Waterfront?* may consist of a simple *confirmation* (e.g. *yes*), a simple *rejection* (e.g. *no*), a *correction* (e.g. *no, Braddock avenue*) or a *topic change* (e.g. *no, leave at 7* or *yes, and go to Oakland*). Table 3 contains more examples of post-confirmation user utterances to the *Let's Go!* system. The user's response type has implications for further system processing. In particular, because *Let's Go!* uses state-specific language models, corrections and topic changes are more likely to be misrecognized, leading to cascading errors and negatively affecting task completion rates and user satisfaction.

In our analysis of *Let's Go!* data from 2005, users specify a concept in 18% of post-confirmation utterances. 15.6% of post-confirmation utterances in the 2005 dataset contain a *place* concept, 3.2%

5. A bus route can also be automatically inferred by the system from the user's start and destination locations.

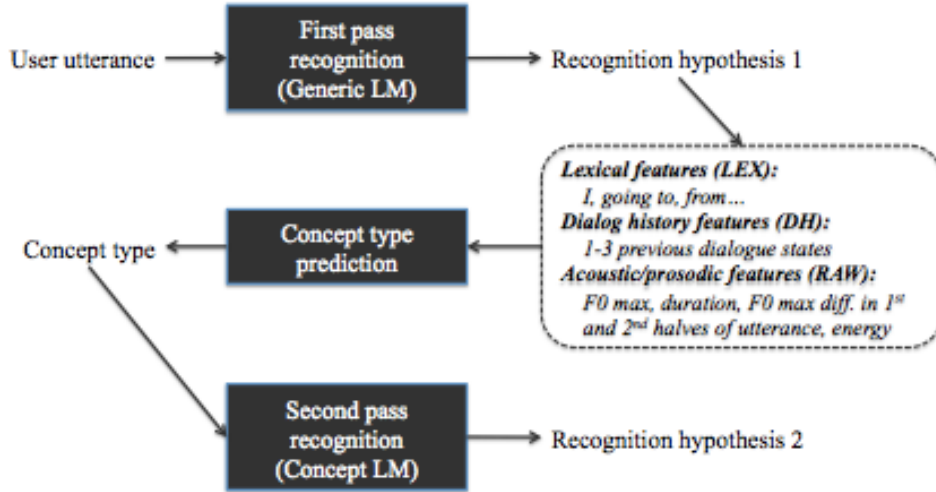


Figure 5: Two-pass automatic speech recognition

contain a *time* concept, and 6.4% contain a *bus* concept (see *Concept Type Features* in Table 4). Because utterances with a concept are not well represented in the *confirm* language model, recognition is likely to fail on utterances containing a concept. Even though such utterances are relatively infrequent, they are disproportionately important. As Figure 4 shows, in *Let's Go!* the word error rate on post-confirmation *Let's Go!* utterances containing a concept is 10% higher than on utterances without a concept. Our previous analysis of the Communicator corpus (Walker et al. 2002) shows that the probability of a consecutive error (when a sequence of utterances is misrecognized) is significantly higher than the probability of an initial error (Stoyanchev 2009). Correct prediction of the content of post-confirmation user utterances can lead to improved speech recognition, fewer and shorter sequences of speech recognition errors, and improved dialogue system performance. In short, post-confirmation utterances present an opportunity for responsive adaptation that is likely to have a positive impact on dialogue success rates.

5. Experimental Approach

5.1 Two-pass Speech Recognition

We adopt the two-pass recognition architecture previously introduced by Young (1994) and illustrated in Figure 5. In the first pass, the input utterance is processed using the *confirm* language model. Recognition may fail on concept words such as *Oakland* or *61C*, but is likely to succeed on closed-class words (e.g. *yes*, *no*). Then, the concept type predictor uses acoustic, lexical and dialogue history features to determine the task-related *concept type(s)* likely to be present in the utterance. In the second recognition pass, any utterance containing a concept type is re-processed using a concept-specific language model.

| Event | 2005 | | 2006 | |
|--------------------------------------|-------|-------|-------|-------|
| | num | % | num | % |
| Total dialogues | 2411 | | 1430 | |
| Total confirm utts | 9098 | 100 | 9028 | 100 |
| Confirms utts with a concept | 2194 | 24 | 1635 | 18.1 |
| Dialogue State | | | | |
| Total confirm place system utts | 5548 | 61 | 5347 | 59.2 |
| Total confirm bus system utts | 1763 | 19.4 | 1589 | 17.6 |
| Total confirm time system utts | 1787 | 19.6 | 2011 | 22.3 |
| Concept Type Features | | | | |
| User's post-confirm utts with place | 1416 | 15.6 | 1007 | 11.2 |
| User's post-confirm utts with time | 296 | 3.2 | 305 | 3.4 |
| User's post-confirm utts with bus | 584 | 6.4 | 323 | 3.6 |
| Lexical Features | | | | |
| User's post-confirm utts with 'yes' | 4395 | 48.3 | 3693 | 40.9 |
| User's post-confirm utts with 'no' | 2076 | 22.8 | 1564 | 17.3 |
| User's post-confirm utts with 'I' | 203 | 2.2 | 129 | 1.4 |
| User's post-confirm utts with 'from' | 114 | 1.3 | 185 | 2.1 |
| User's post-confirm utts with 'to' | 204 | 2.2 | 237 | 2.6 |
| Acoustic/Prosodic Features | | | | |
| feature | mean | stdev | mean | stdev |
| Duration (seconds) | 1.341 | 1.097 | 1.365 | 1.242 |
| Energy (RMS mean) | 0.037 | 0.033 | 0.055 | 0.049 |
| F0 mean | 183.0 | 60.86 | 185.7 | 58.63 |
| F0 max | 289.8 | 148.5 | 296.9 | 146.5 |

Table 4: Statistics on post-confirmation utterances

5.2 Experimental Data

The data we used for concept type prediction and language model adaptation comes from the first two months of *Let's Go!* system operation in 2005 (2411 dialogues), and one month in 2006 (1430 dialogues). Researchers at Carnegie Mellon transcribed this data and labeled the presence of task-relevant concepts by hand. In the annotated transcripts, the following concepts are labeled: *neighborhood*, *place*, *time*, *hour*, *minute*, *time-of-day*, and *bus*. We collapsed these concepts into three concept types: *time*, (including *time*, *hour*, *minute*, *part of the day*, and *day of the week*), *place* (including *place* and *neighborhood*), and *bus* (see Table 2).

Table 4 shows statistics on post-confirmation user's utterances in *Let's Go!* for the 2005 and 2006 datasets. Perhaps because of system improvements and user experience, the two data sets have some interesting differences. Most confirmation prompts in both data sets are for a *place* (61% and 59.2% respectively). However, in the 2005 dataset the system prompted for *bus* concept confirmation more often than in the 2006 dataset (19.4% vs. 17.6%). Perhaps some users figured out that *bus* is not a required piece of information; start and end locations are sufficient for the system to figure out the bus route.

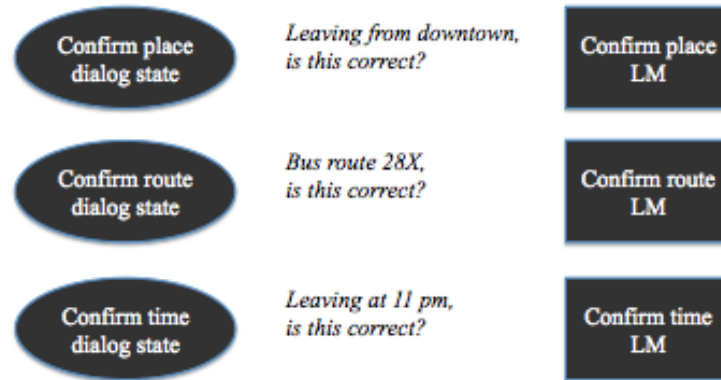


Figure 6: A confirm-type baseline approach to language modeling

There are also differences in user responses to confirmation prompts. The proportion of responses containing *yes*, *no*, and/or a concept all dropped between 2005 and 2006. This may be caused by users in the 2006 dataset using more variation when responding to confirmation prompts.

We also observe some differences in duration of user utterances between the two datasets. This may be due to improvement in automatic detection of the end of user speech. Finally, the 2006 dataset shows higher energy RMS mean measure; this may be due to change in the hardware settings.

Because of these differences between the two datasets, we used only the 2006 dataset for the concept type classification experiments. We used the 2005 dataset to build language models, as was done in the live version of the *Let's Go!* system that was used to collect the 2006 dataset. In our speech recognition experiment, only the language models used for recognizing post-confirmation user utterances are different from the language models used in the 2006 system.

6. Predicting Concept Type Experiment

In this section we describe our experiments on concept type prediction. First, we describe three methods for concept type prediction, two baseline methods and a machine learning method. Second, we present experimental results comparing our machine learning method to the two baseline methods, and comparing the performance of different feature sets for concept type prediction. We present experimental results for both transcribed speech (best possible performance) and automatically recognized speech (real-world performance). In Section 7.2 we present speech recognition results for two-pass speech recognition using the three concept type prediction methods.

6.1 Baseline Methods

6.1.1 NO-CONCEPT BASELINE PREDICTION

Our first baseline method predicts *no concept* for all utterances. Since the majority of post-confirmation utterances in our data do not contain a concept, the overall accuracy of this prediction method is 82%. However, it is not useful for improving speech recognition on utterances containing a concept as its prediction for these utterances is always incorrect.

| | System's confirm state | | |
|---------------------|------------------------|------|------|
| | place | bus | time |
| 2005 dataset | | | |
| user confirm_place | 0.86 | 0.13 | 0.01 |
| user confirm_bus | 0.18 | 0.81 | 0.01 |
| user confirm_time | 0.07 | 0.01 | 0.92 |
| 2006 dataset | | | |
| user confirm_place | 0.87 | 0.10 | 0.03 |
| user confirm_bus | 0.34 | 0.64 | 0.02 |
| user confirm_time | 0.15 | 0.13 | 0.71 |

Table 5: System's confirmation state vs. user concept type

6.1.2 CONFIRM-TYPE BASELINE PREDICTION

Our second baseline method uses the concept type being confirmed by the system (Figure 6) as the concept type for the input user utterance. For example, if the system requests confirmation of a *place*, this method predicts that the user's post-confirmation utterance will contain a *place* concept.

There are two problems with the second baseline approach. First, the majority of utterances (82% in the 2006 dataset) do not contain any concept, so the overall accuracy of this prediction method is less than 18%. Second, users may attempt topic changes in post-confirmation utterances using a different concept than the one confirmed. Table 5 shows a confusion matrix for confirmation prompt concept type and post-confirmation utterance concept type. For example, in the 2006 dataset, after a system confirmation prompt for a *bus*, a *bus* concept is used in only 64% of concept-containing user utterances.

6.1.3 MACHINE LEARNING METHOD

We use decision trees to classify each post-confirmation user utterance according to the concept type(s) it contains (*place*, *time*, *bus* or *none*). We experimented with using the system confirmation concept type feature (the same one used in the second baseline method described above), lexical features, prosodic features, and dialogue history features. All of these features (except for those derived from the transcript) are available at run-time and can be used in a live system. The concept prediction performance on lexical features from transcripts is used to estimate the best possible performance of the lexical feature set. The concept prediction performance on lexical features from the output of the first-pass recognition is the realistic performance given noisy input. The features are outlined in Table 6 and described below.

System Confirm-Type Feature (DIA) The system confirm-type feature is the same one used in the *confirm-type* baseline method. It indicates the concept type requested in the confirmation prompt and takes the values *place*, *bus*, or *time*.

Acoustic Features (RAW) Acoustic features are extracted from the raw audio of the user utterances. We use utterance maximum pitch (F0 max), energy (RMS), duration, and the difference between F0 max in the first and second halves of the utterance. We selected these features based on the work of Litman et al. (2006) on detecting speech recognition errors; we anticipated that these features would help distinguish corrections and rejections from confirmations. We used Pratt (Boersma and Weenink) scripts to automatically extract these features from the audio.

| Feature type | Feature source | Feature description |
|------------------------------------|--|--|
| System confirm-type (DIA) | system log | System’s confirmation prompt concept type (<i>confirm_time</i> , <i>confirm_place</i> , or <i>confirm_bus</i>) |
| Acoustic (RAW) | raw speech | F0 max; RMS max; RMS mean; Duration; Difference between F0 max in first half and in second half |
| Dialogue history (DH1, DH3) | 1-3 previous utterances | System’s dialogue states of previous utterances (<i>first_query</i> , <i>place</i> , <i>time</i> , <i>confirm_place</i> , <i>confirm_time</i> , or <i>confirm_bus</i>); [transcribed speech only] Concept(s) that occurred in user’s utterances (YES/NO for each of the concepts <i>place</i> , <i>bus</i> , <i>time</i>) |
| Lexical (LEX) | transcript/first-pass recognition output | Presence of specific lexical items; Number of tokens in utterance; [transcribed speech only] String edit distance between current and previous user utterances |
| ASR confidence score (ASR) | first-pass recognition output | Speech recognizer confidence score |
| Concept type match (CTM) | first-pass recognition output | Presence of concept-specific lexical items |

Table 6: Features for concept type prediction

Dialogue History Features (DH) We use either one or three utterances of dialogue history (DH1, DH3). These features capture information about the dialogue state history (SH) and concept history (CH). In DH1, the system dialogue state for, and the concept type(s) present in, the previous user utterance are recorded. In DH3, the system dialogue states for, and the concept types presented in, the three previous user utterances are recorded. The system dialogue state values can be *first_query*, *place*, *time*, *confirm_place*, *confirm_time*, or *confirm_bus*. This feature is extracted from the system log. The concept history features are extracted from the annotated, transcribed user speech and are represented as triples of binary values indicating the presence of *place*, *time*, or *bus* concepts in the user’s utterance. Table 7 shows the values of dialogue state (DIA), state history, and concept history features on an extract from a *Let’s Go!* dialogue. User utterance #4 is a response to a confirmation prompt about a bus route number, so its DIA feature value is *confirm_bus*. The value for SH1 is *first_query*, the value of the preceding system state corresponding to system utterance #1. The value for CH1 contains *bus* because the previous user utterance (#2) mentioned a bus. For utterance #4 there are no user utterances more than one back, so SH2, SH3, CH2, and CH3 are undefined.

Lexical Features (LEX) LEX features include non-concept-value words and bigrams from the user’s current utterance, such as *go*, *leave*, *to*, or *from*. These features can be highly indicative of the presence or absence of any concept type, as well as of the presence of a particular concept type. For example, *going to* may be highly correlated with a *place* concept and *leaving at* may be correlated with a *time* concept. We explored two methods for identifying the most salient lexical features: manual identification and mutual information-based identification. Both of these methods select a set of salient lexical features that are then used for concept type prediction.

- **Manual approach:** We manually selected five lexical features: *yes* (indicates a confirmation), *no* (indicates a rejection), *to* and *from* (indicate presence of concept types *place* and *time*), and *!* (indicates a complete sentence). These features were selected based on a heuristic estimate of their importance and their high relative frequency in the corpus.

| # | Speaker | Utterance | DIA | State history | Concept history |
|---|-----------------|--|------------|---|--|
| 1 | S (first query) | What can I do for you? | | | |
| 2 | U | I want to catch the 28x | | | |
| 3 | S (conf) | The 28X. Did I get that right? | | | |
| 4 | U (post conf) | Yes. From the airport to down-town | conf bus | SH1=first_query SH2= \emptyset SH3= \emptyset | CH1=bus CH2= \emptyset CH3= \emptyset |
| 5 | S (conf) | Leaving from the Airport. Is this correct? | | | |
| 6 | U (post conf) | Yes. | conf place | SH1=confirm_bus SH2=first_query SH3= \emptyset | CH1=place CH2=bus CH3= \emptyset |
| 7 | S (conf) | Okay. Going to Downtown. Is this correct? | | | |
| 8 | U (post conf) | Yes. | conf place | SH1=confirm_place SH2=confirm_bus SH3=first_query | CH1= \emptyset CH2=place CH3=bus |

Table 7: Dialogue state and history features example

- **Mutual information approach:** This method was successfully used by Gorin et al. (1997) in a call routing system for detecting salient phrases. We used it to select lexical features according to the *mutual information* between potential feature and concept types (Manning et al. 2008). We extracted lexical features (unigrams and bigrams) from the transcribed user utterances. We removed all words that realize concept values (e.g. *61C*, *Squirrel Hill*), as these are likely to be misrecognized in the first pass recognition of a post-confirmation utterance. We then computed the mutual information between each remaining potential lexical feature and each concept type, and selected the features with the highest mutual information scores.

We computed the mutual information score I for each lexical feature t and each concept type class $c \in \{place +, place -, time +, time -, bus +, bus -\}$ as follows:

$$I = \frac{N_{tc}}{N} * \log_2 \frac{N * N_{tc}}{N_{t.} * N_{.c}} + \frac{N_{0c}}{N} * \log_2 \frac{N * N_{0c}}{N_{0.} * N_{.c}} + \frac{N_{t0}}{N} * \log_2 \frac{N * N_{t0}}{N_{t.} * N_{.0}} + \frac{N_{00}}{N} * \log_2 \frac{N * N_{00}}{N_{0.} * N_{.0}}$$

where N_{tc} = number of utterances where t co-occurs with c , N_{0c} = number of utterances with c but without t , N_{t0} = number of utterances where t occurs without c , N_{00} = number of utterances with neither t nor c , $N_{t.}$ = total number of utterances containing t , $N_{.c}$ = total number of utterances containing c , and N = total number of utterances.

Table 8 shows several lexical features with high mutual information for each concept type. For example, the feature *to* co-occurs with the concept *place* in 217 utterances (N_{tc}), and occurs without the concept *place* in only 39 utterances (N_{t0}), so presence of this feature in an utterance is indicative of presence of a *place*. The feature *yes*, on the other hand, occurs without the concept *place* in 3652 utterances and with the concept *place* in only 41 utterances, so it is indicative of absence of *place*.

| Features | N_{0c} | N_{t0} | N_{00} | N_{tc} | Info. measure |
|--------------|----------|----------|----------|----------|------------------|
| place | | | | | |
| yes | 964 | 3652 | 2501 | 41 | 0.127 |
| to | 788 | 39 | 6114 | 217 | 0.069 |
| from | 828 | 25 | 6128 | 177 | 0.058 |
| going | 891 | 14 | 6139 | 114 | 0.038 |
| bus | | | | | |
| yes | 307 | 3678 | 3158 | 15 | 0.036 |
| the | 232 | 80 | 6756 | 90 | 0.036 |
| the next | 297 | 26 | 6810 | 25 | 0.0089 |
| time | | | | | |
| yes | 167 | 3690 | 3298 | 3 | 0.022 |
| at | 151 | 26 | 6962 | 19 | 0.0085 |
| on | 166 | 23 | 6965 | 4 | 0.0008 |

Table 8: Mutual information for selected features.

We tried two methods for selecting features with the highest mutual information. In the first method, we selected for each concept type the 50 features with the highest mutual information. In the second method, we selected for each concept type the 30 features with the highest mutual information that occurred at least 20 times in the training data⁶.

ASR Confidence Score Feature (ASR) We used the speech recognizer’s confidence score for the first-pass recognition for each utterance (using the *generic-confirm* language model).

Concept Type Match Features (CTM) The CTM features indicate whether a user’s utterance matches a concept value for a particular concept type. We tokenized all concept values (names of bus stops, places, buses, and times). Each automatically recognized user utterance was matched to the bag of tokens for each of the concepts, and the result assigned to one of the three binary features *CTM_place*, *CTM_bus*, and *CTM_time*. For example, the *CTM_place* feature is set to **true** when a recognized utterance matches a part of one of the *place* concept values, such as *street* or *avenue*.

For transcribed speech there is a one-to-one correspondence between presence of the concept and the CTM feature, so this feature alone gives 100% concept prediction accuracy. Consequently, we only evaluate this feature for recognized speech. We hypothesize that the CTM feature will improve cases where part of (but not the whole) concept value is recognized in first-pass recognition. So, if in the utterance *Madison avenue*, *avenue* (but not *Madison*), is recognized in first-pass recognition, the CTM feature can flag the utterance for *place*, helping the classifier to correctly assign the *place* type to the utterance. Then, in second-pass recognition the utterance will be decoded with a *place* concept-specific language model, potentially improving speech recognition performance.

6.2 Experimental Results

In this section we present experimental results for concept type prediction for both baseline methods and for the machine learning method. We performed a series of 10-fold cross-validation experiments

6. We aimed to select an equal number of features for each concept type, while ensuring that each feature had mutual information in the top 25%. 30 was an empirically derived threshold for the number of lexical features to satisfy these conditions.

| Measure | Description | Formula |
|----------------|---|---------------------------------------|
| <i>pre+</i> | precision of predicting presence of a concept | $tp/(tp+fp)$ |
| <i>rec+</i> | recall of predicting presence of a concept | $tp/(tp+fn)$ |
| <i>f+</i> | f-measure for predicting presence of a concept | $2*[rec+]*[pre+] / ([pre+] + [rec+])$ |
| <i>acc</i> | overall accuracy | $(tp+tn)/(tp+tn+fp+fn)$ |
| <i>switch+</i> | error due to misclassification of utts with concept with an incorrect concept | $1-(tp/all\ utts\ with\ concept)$ |
| <i>switch</i> | error due to misclassification of any utt with an incorrect concept | $1-((tp+fp)/all\ utts)$ |

Table 9: Measures of concept prediction. tp=true positives, tn=true negatives, fp=false positives, fn=false negatives

to examine the impact on concept type prediction of different methods and of different feature combinations. We trained three binary classifiers for each experiment, one for each concept type, i.e. we separately classified each post-confirmation utterance as *place +* or *place -*, *time +* or *time -*, and *bus +* or *bus -*. We used Weka’s implementation of the J48 decision tree classifier (Witten and Eibe 2005).⁷

We report overall classification performance separately for feature combinations using lexical features from transcribed speech (Table 11) and from automatically recognized speech (Table 13). Performance for each concept type is reported in Table 12 for transcribed speech and Table 14 for recognized speech. The results on transcribed speech give us an idea of the best possible performance on concept type classification. The results on recognized speech provide a realistic estimate of the performance in a live dialogue system.

Table 9 outlines each of our performance measures and describes how they are computed. For each experiment, we report precision (*pre+*) and recall (*rec+*) for determining *presence* of each concept type, and overall classification accuracy for each concept type (*place*, *bus* and *time*). We do not report precision or recall for determining *absence* of each concept type. Because in the data 82.2% of utterances do not contain any concepts (see Table 4), precision and recall for determining absence of each concept type are above .9 in each of the experiments. We also report overall *pre+*, *rec+*, f-measure (*f+*), and classification accuracy across the three concept types.

To get an estimate of the potential impact on speech recognition performance, we also report the percentage of *switch+* errors and *switch* errors. *Switch+* errors are the proportion of utterances containing a concept c_A that are classified as containing a different concept c_B . Utterances containing *bus* classified incorrectly as *time/place*, *time* as *bus/place*, and *place* as *bus/time* are counted as *switch+* errors. In the second pass of speech recognition these utterances will be decoded with a language model built for a concept different from the concept in the utterance and will be likely to have a higher word error rate. The *switch* error rate is the proportion of all utterances misclassified

7. We used decision trees because they gave good performance on our data set compared with other classification methods, and because they permit examination of the features in the learned models.

| Features | Classification accuracy | |
|-------------------------------|-------------------------|-------------|
| | rec+ | acc |
| LEX _{manual5} | 0.55 | 0.89 |
| LEX _{topMI50} | 0.52 | 0.88 |
| LEX _{freq30} | 0.56 | 0.89 |
| RAW+DH+LEX _{manual5} | 0.57 | 0.89 |
| RAW+DH+LEX ₅₀ | 0.56 | 0.89 |
| RAW+DH+LEX _{freq30} | 0.62 | 0.90 |

Table 10: Comparing approaches to selection of lexical features. Concept type classification accuracy is reported on lexical features from recognized speech. Best overall values in each group are highlighted in bold.

as containing one of the concepts. *Switch* errors include all of the *switch+* errors and also errors on utterances with no concept classified as *place*, *bus* or *time*.

Utterances classified as containing one of the three concept types are subject to second-pass recognition using a concept-specific language model. Utterances that are classified correctly as containing a particular concept type (*rec+* represents proportion of correctly classified utterances with a concept) will be subject to second-pass recognition using a more appropriate language model. Speech recognition performance on these utterances may improve in the second pass of speech recognition. On the other hand, utterances that are incorrectly classified as containing a particular concept type (*switch+*) will be subject to second-pass recognition using a poorly-chosen language model. This is a severe error that is likely to cause speech recognition performance to suffer. This means that we want to maximize *rec+* and minimize *switch+* errors.

6.2.1 PERFORMANCE OF BASELINE METHODS

The **No-Concept** baseline achieves overall classification accuracy of 82% but *rec+* of 0 (see Table 11). *Switch+* on the **No-Concept** baseline is 0 because all utterances are classified as ‘no concept’. Misclassifications of utterances with a concept as ‘no concept’ are not counted as *Switch+* errors. (Utterances with a concept misclassified as *none* will be decoded with the same *generic confirm* language model in the second pass of recognition. The word error rate from second-pass recognition will be the same as from first-pass recognition.)

The **Confirm-type** baseline achieves *rec+* of .79, but overall classification accuracy of only 14%. *Switch+* is .17.

6.2.2 PERFORMANCE OF MACHINE LEARNING METHOD

In this section, we explore the impact of different feature sets on performance of the machine learning method. All results are averaged 10-fold cross-validation results, and all models use the DIA feature. For simplicity, we will call the model trained on LEX features the *LEX model*, the model trained on RAW features, the *RAW model*, and so on. We determine significance of the difference between conditions using the inference on proportions test with Bonferroni correction for *rec+* (which is the proportion of utterances with concepts that were correctly classified).

| Features | Overall | | | | | |
|---------------------------------|-------------|-------------|-------------|-------------|------------------|-----------------|
| | pre+ | rec+ | f+ | acc | switch+ error | switch error |
| No Concept baseline | 0 | 0 | 0 | 0.82 | 0 | 0 |
| Confirm-type baseline | 0.14 | 0.79 | 0.24 | 0.14 | 0.170 | 0.723 |
| Features from current utterance | | | | | | |
| RAW | 0.67 | 0.34 | 0.45 | 0.85 | 0.064 | 0.040 |
| LEX | 0.87 | 0.72 | 0.79 | 0.93 | 0.073 | 0.032 |
| LEX_RAW | 0.88 | 0.70 | 0.78 | 0.93 | 0.074 | 0.030 |
| +Features from dialogue history | | | | | | |
| DH1_LEX | 0.88 | 0.81 | 0.84 | 0.95 | 0.055 | 0.029 |
| DH3_LEX | 0.89 | 0.78 | 0.83 | 0.94 | 0.052 | 0.026 |

Table 11: Overall concept type classification results: transcribed speech (all models include feature DIA). Best overall values in each group are highlighted in bold.

| Features | Place | | | Time | | | Bus | | |
|---------------------------------|-------|------|------|------|------|------|------|------|------|
| | pre+ | rec+ | acc | pre+ | rec+ | acc | pre+ | rec+ | acc |
| No Concept baseline | 0 | 0 | .86 | 0 | 0 | 0.81 | 0 | 0 | .92 |
| Confirm-type baseline | 0.87 | 0.85 | 0.86 | 0.64 | 0.54 | 0.58 | 0.71 | 0.87 | 0.78 |
| Features from current utterance | | | | | | | | | |
| RAW | 0.65 | 0.53 | 0.92 | 0.25 | 0.01 | 0.96 | 0.38 | 0.07 | 0.96 |
| LEX | 0.81 | 0.88 | 0.96 | 0.77 | 0.48 | 0.98 | 0.83 | 0.59 | 0.98 |
| LEX_RAW | 0.83 | 0.84 | 0.96 | 0.75 | 0.54 | 0.98 | 0.76 | 0.59 | 0.98 |
| +Features from dialogue history | | | | | | | | | |
| DH1_LEX | 0.85 | 0.91 | 0.97 | 0.72 | 0.63 | 0.98 | 0.89 | 0.83 | 0.99 |
| DH3_LEX | 0.85 | 0.87 | 0.97 | 0.72 | 0.59 | 0.98 | 0.92 | 0.82 | 0.99 |

Table 12: Concept type classification results for each concept: transcribed speech (all models include feature DIA).

Lexical Feature Selection Approaches We compare the performance of the machine learning method using three approaches to selecting lexical features: (a) manual selection, (b) LEX_{50} , automatically selecting the 50 features with the highest mutual information; and (c) LEX_{freq30} , automatically selecting the 30 features with the highest mutual information that occur at least 20 times in the training data.

As Table 10 shows, the LEX_{freq30} feature set achieves the highest classification accuracy and *rec+*, both on its own and combination with other feature sets. The prosodic (RAW) and dialogue history (DH) feature sets lead to additional improvements in performance. Therefore, in the experiments described later in this section, all LEX features are selected using the LEX_{freq30} approach.⁸

Features from the Current Utterance (RAW, LEX, LEX_RAW) We look at the performance of simple models using only the dialogue state (DIA) feature, with lexical (LEX) and acoustic/prosodic (RAW) features from the current utterance. A model trained on RAW features alone achieves *rec+*

8. Switch+ errors are not reported here as they did not differ across the lexical feature selection approaches.

| Features | Overall | | | | | |
|---|-------------|-------------|-------------|-------------|------------------|-----------------|
| | pre+ | rec+ | f+ | acc | switch+ error | switch error |
| No Concept baseline | 0 | 0 | 0 | 0.82 | 0 | 0 |
| Confirm-type baseline | 0.14 | 0.79 | 0.24 | 0.14 | 0.170 | 0.723 |
| Features from current utterance | | | | | | |
| RAW | 0.67 | 0.34 | 0.45 | 0.85 | 0.064 | 0.040 |
| LEX | 0.75 | 0.56 | 0.64 | 0.89 | 0.099 | 0.049 |
| LEX_RAW | 0.76 | 0.60 | 0.67 | 0.90 | 0.103 | 0.051 |
| +Features from dialogue history | | | | | | |
| DH1_LEX_RAW | 0.77 | 0.60 | 0.67 | 0.90 | 0.082 | 0.046 |
| DH3_LEX_RAW | 0.77 | 0.62 | 0.68 | 0.90 | 0.072 | 0.046 |
| +Features specific to recognized speech | | | | | | |
| ASR_DH3_LEX_RAW | 0.77 | 0.62 | 0.68 | 0.90 | 0.072 | 0.045 |
| CTM_DH3_LEX_RAW | 0.85 | 0.74 | 0.79 | 0.93 | 0.039 | 0.029 |
| CTM_ASR_DH3_LEX_RAW | 0.85 | 0.74 | 0.79 | 0.93 | 0.042 | 0.030 |

Table 13: Overall concept type classification results: recognized speech (all models include feature DIA). Best overall values in each group are highlighted in bold.

| Features | Place | | | Time | | | Bus | | |
|---|-------|------|------|------|------|------|------|------|------|
| | pre+ | rec+ | acc | pre+ | rec+ | acc | pre+ | rec+ | acc |
| No Concept baseline | 0 | 0 | .86 | 0 | 0 | 0.81 | 0 | 0 | .92 |
| Confirm-type baseline | 0.87 | 0.85 | 0.86 | 0.64 | 0.54 | 0.58 | 0.71 | 0.87 | 0.78 |
| Features from current utterance | | | | | | | | | |
| RAW | 0.65 | 0.53 | 0.92 | 0.25 | 0.01 | 0.96 | 0.38 | 0.07 | 0.96 |
| LEX | 0.70 | 0.70 | 0.93 | 0.67 | 0.15 | 0.97 | 0.65 | 0.62 | 0.98 |
| LEX_RAW | 0.70 | 0.72 | 0.93 | 0.66 | 0.38 | 0.97 | 0.68 | 0.57 | 0.98 |
| +Features from dialogue history | | | | | | | | | |
| DH1_LEX_RAW | 0.71 | 0.68 | 0.93 | 0.68 | 0.38 | 0.97 | 0.78 | 0.63 | 0.98 |
| DH3_LEX_RAW | 0.71 | 0.70 | 0.93 | 0.67 | 0.42 | 0.97 | 0.79 | 0.63 | 0.98 |
| +Features specific to recognized speech | | | | | | | | | |
| ASR_DH3_LEX_RAW | 0.71 | 0.70 | 0.93 | 0.69 | 0.42 | 0.97 | 0.79 | 0.63 | 0.98 |
| CTM_DH3_LEX_RAW | 0.82 | 0.82 | 0.96 | 0.86 | 0.71 | 0.99 | 0.76 | 0.68 | 0.98 |
| CTM_ASR_DH3_LEX_RAW | 0.82 | 0.81 | 0.96 | 0.86 | 0.69 | 0.99 | 0.76 | 0.68 | 0.98 |

Table 14: Concept type classification results for each concept type: recognized speech (all models include feature DIA).

| Concept type | Average # non-concept words in utt | Average # words in concept | Average # chars in concept |
|--------------|------------------------------------|----------------------------|----------------------------|
| place | 1.29 | 2.2 | 12.8 |
| bus | 1.63 | 2.9 | 10 |
| time | 1.73 | 1.7 | 6.6 |

Table 15: Length of user utterances containing concept

of 0.34 and overall accuracy of 0.85 (see Table 11). This model performs surprisingly well, beating both baselines in overall accuracy (0.85 vs. 0.82 & 0.14 for the *no-concept* & *confirm-type* baselines, both differences significant at $p < .001$). However, this model only works well for *place* concepts. As shown in Table 14, the *rec+* for the RAW model is 0.53 for the *place* concept, but only 0.01 and 0.07 for the *time* and *bus* concepts. This result indicates that utterances containing values for the *place* concept, but not the *time* or *bus* concepts, contain prosodic information that can be used for determining presence of a concept.

One possible reason for this difference in performance may be the lack of training data for the *time* and *bus* concepts (see Table 4). Another reason may be the difference in duration of the concept values. Table 15 shows the average number of non-concept words in an utterance, the average number of words in a concept value in an utterance, and the average number of characters in a concept value in an utterance⁹. Realizations of values for the *time* concept are much shorter than realizations of values for the *place* and *bus* concepts¹⁰.

The LEX model for both transcribed (Table 11) and recognized (Table 13) speech achieves significantly higher *rec+* than the RAW model (0.72 & 0.56 vs. 0.34) and overall accuracy (0.93 & 0.89 vs. 0.85, all differences significant at $p < .001$). Despite higher *rec+*, for recognized speech, the LEX model has significantly more *switch+* errors than the RAW model (0.099 vs 0.064, $p < .001$). This means that LEX and RAW models differ in the type of errors that they make. The LEX model makes more errors that involve mislabeling utterances with a concept by a different concept while the RAW model makes more errors by mislabeling an utterance with a concept as *no concept* or an utterance without a concept as containing a concept. This suggests that combining the two models may improve the performance of concept prediction.

For transcribed speech, the LEX_RAW model does not perform significantly differently from the LEX model in terms of overall accuracy, *rec+*, or *switch+* errors. However, for recognized speech, LEX_RAW achieves significantly higher *rec+* (0.60) and overall accuracy (0.90) than LEX (*rec+* 0.56 and *acc* 0.89, $p < .001$). Lexical features from transcribed speech are very good indicators of concept type. However, lexical features from recognized speech are noisy, so concept type classification for recognition output can be improved by using acoustic/prosodic (RAW) features.

Prediction accuracy varies widely across concept types. Figure 7 depicts *rec+* for *place*, *time*, and *bus* concepts using LEX from transcribed speech, LEX from recognized speech, and LEX_RAW from recognized speech. We achieve highest *rec+* for the *place* concept for each of the feature combinations. This may be partially due to the fact that we have more training data for the *place* concept than for the other concepts, and partially due to more informative lexical features (e.g. *to*, *from*) in utterances containing values for the *place* concept. The *time* concept has the lowest *rec+*, and the biggest drop in performance due to recognition errors (difference between LEX on

9. We use the number of characters to approximate the number of syllables.

10. The most common value for the *time* concept, *now*, is 3 characters long.

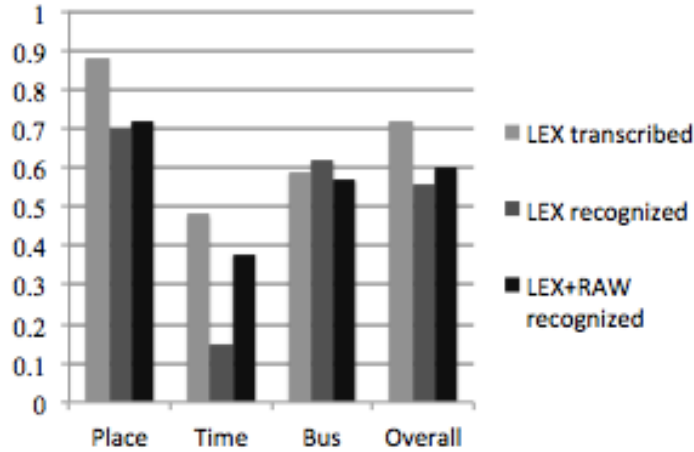


Figure 7: Value of *rec+* for each concept (graphical illustration of the results in Tables 12 and 14)

transcribed and LEX on recognized speech). However, prosodic features have the biggest impact on *rec+* for the *time* concept, improving *rec+* from a low 0.15 to 0.38.

Overall, models containing only features from the current utterance perform significantly worse than the *confirmation state* baseline in terms of *rec+* ($p < .001$). However, they have significantly higher accuracy and fewer *switch+* errors ($p < .001$).

Features from the Dialogue History (DH1, DH3) Next, we add features from the dialogue history to our best-performing models so far. For transcribed speech (Table 11), a model with one utterance of history (DH1_LEX) performs significantly better than LEX in terms of *rec+* (0.81 vs 0.72), overall accuracy (0.95 vs. 0.93), and *switch+* errors (0.055 vs. 0.073, $p < .001$). A model with three utterances of history (DH3_LEX) performs significantly worse than DH1_LEX in terms of *rec+* (0.78 vs. 0.81 $p < 0.05$). For recognized speech (Table 13), neither DH1_LEX_RAW nor DH3_LEX_RAW is significantly different from LEX_RAW in terms of *rec+* or overall accuracy. However, both DH1_LEX_RAW and DH3_LEX_RAW do perform significantly better than LEX_RAW in terms of *switch+* errors (.082 and .072 vs. .103, $p < .05$). There are no significant performance differences between DH1_LEX_RAW and DH3_LEX_RAW.

Features Specific to Recognized Speech (ASR, CTM) Finally, we add the speech recognition confidence score (ASR) and concept type match (CTM) features to models trained on recognized speech. We hypothesized that the classifier can use the recognizer’s confidence score to decide whether an utterance is likely to have been misrecognized. However, ASR_DH3_LEX_RAW is not significantly different from DH3_LEX_RAW in terms of *rec+*, overall accuracy or *switch+* errors. This agrees with the findings of Lemon and Konstantas (2009) that ASR confidence scores show lower information gain than other features when classifying recognition hypothesis quality.

By contrast, adding the concept type match (CTM) feature to DH3_LEX_RAW leads to a large improvement on all measures: a 12% absolute increase in *rec+* (from .62 to .74), a 3% absolute increase in overall accuracy (from .90 to .93), and decreases in *switch+* errors (from .072 to .042), all statistically significant at $p < .001$. There are no statistically significant differences between CTM_DH3_LEX_RAW and CTM_ASR_DH3_LEX_RAW.

6.2.3 SUMMARY AND DISCUSSION

In this section we evaluated different models for concept type prediction. The best performing transcribed speech model, DH1_LEX, significantly outperforms the Confirm-type baseline on overall accuracy and on *switch+* and *switch* errors ($p < .001$), and is not significantly different on *rec+*. The best performing recognized speech model, CTM_DH3_LEX_RAW, significantly outperforms the Confirm-type baseline on overall accuracy and on *switch+* and *switch* errors, but is significantly worse on *rec+* ($p < .001$). The best transcribed speech model achieves significantly higher *rec+* and overall accuracy than the best recognized speech model ($p < .01$), but is not significantly different in terms of *switch+* errors.

Although the best-performing concept type prediction models achieve very high accuracy, high *rec+* rates, and low rates of *switch+* errors, speech recognition is a noisy process. Therefore, in order to confirm that concept type prediction can lead to improved speech recognition performance, we ran a speech recognition experiment.

7. Speech Recognition Experiment

In this section we look at the impact of concept type prediction on speech recognition performance in *Let's Go!* data. We hypothesized that speech recognition performance for post-confirmation utterances containing a concept can be improved with the use of concept-specific language models. In this set of experiments, we (1) compare the existing generic *confirm* language model used in *Let's Go!* with the proposed *concept-specific* language model adaptation strategy; (2) compare two methods for selecting user utterances for building language models; and (3) evaluate the impact of different methods of concept type prediction on *concept-specific* language model adaptation.

7.1 Method

We used the PocketSphinx speech recognition engine (Huggins-Daines et al. 2006) with gender-specific telephone-quality acoustic models built for Communicator (Rudnicky et al. 2000). We trained trigram language models using 0.5 ratio discounting with the CMU language modeling toolkit (Xu and Rudnicky 2000)¹¹. We built state- and concept-specific language models from the *Let's Go!* 2005 data. The language models are hierarchical and encode semantic information (Ward and Issar 1994b), smoothing probabilities for the concepts not used in the data.

We evaluate speech recognition performance on post-confirmation user utterances from the 2006 *Let's Go!* dataset. Each experiment varies in 1) the language model used for the final recognition pass and 2) the method of selecting a language model for use in second-pass recognition.

7.1.1 LANGUAGE MODELS

We used the language model types outlined in Table 16. The *generic-confirm* model is trained on all utterances in the 2005 dataset that were produced in the *confirm* dialogue state. This corresponds to the approach used in the *Let's Go!* 2006 system. The *confirm-type* models are trained using all utterances from the 2005 dataset that were produced in the *confirm* dialogue state following *confirm.place*, *confirm.bus* and *confirm.time* system confirmation prompts respectively. The

11. We used the same speech recognizer, acoustic models, language modeling toolkit, and language model building parameters that were used in the live *Let's Go!* system (see Raux et al. (2005)).

| Concept type prediction method | Language models | Data used for building language models |
|--------------------------------|--|--|
| No-concept | <i>generic-confirm</i> | all post-confirmation utterances |
| Confirm-type | <i>confirm-place</i> <i>confirm-time</i> <i>confirm-bus</i> | post-confirmation utts after <i>confirm.place</i> post-confirmation utts after <i>confirm.time</i> post-confirmation utts after <i>confirm.bus</i> |
| Concept-based | <i>concept-place</i> <i>concept-confirm</i> <i>concept-confirm</i> <i>generic-confirm</i> | post-confirmation utts with <i>place</i> concept post-confirmation utts with <i>time</i> concept post-confirmation utts after <i>bus</i> all post-confirmation utterances |

Table 16: Speech recognition experiment summary: Language models

| Concept type prediction method | Prediction | Decision based on |
|--------------------------------|---|---|
| No-concept | no prediction | |
| Confirm-type | <i>confirm-place</i> <i>confirm-time</i> <i>confirm-bus</i> | post-confirmation utts after <i>confirm.place</i> post-confirmation utts after <i>confirm.time</i> post-confirmation utts after <i>confirm.bus</i> |
| Concept-based | <i>concept-place</i> <i>concept-confirm</i> <i>concept-confirm</i> <i>none</i> | Classifier predicts <i>place</i> concept Classifier predicts <i>time</i> concept Classifier predicts <i>bus</i> Classifier predicts <i>none</i> or multiple concepts |

Table 17: Speech recognition experiment summary: Choosing language models

concept-based models are trained on all utterances from the 2005 dataset that were produced in the *confirm* dialogue state and contain a mention of a *place*, *bus* or *time* respectively.

We used the three methods for choosing language models presented in Section 6 and outlined in Table 17. The first, *no-concept* method simply uses one model for recognizing all utterances. For the second method (*confirm-type*), we use the confirm-type baseline concept type prediction method to choose one of the three confirm-type models. For the third method (*concept-based*), we use one of the classifiers described in the Section 6. The classifier outputs *place*, *time*, *bus*, or *no concept*, and we use the corresponding concept-based model.

7.1.2 RECOGNIZERS

We report results for seven experimental conditions (see Table 18). The experimental conditions vary in method of building and choosing language models. In experimental conditions 1 - 3, recognition was done in a single pass. In the **baseline** experimental condition (1), we used the *generic-confirm* language model to process all post-confirmation utterances. In the **1-pass confirm** experimental condition (2) we used the confirm-type method for building and choosing language models. We built *confirm-place*, *confirm-bus* and *confirm-time* language models to recognize post-confirmation utterances produced following a *confirm.place*, *confirm.bus* and *confirm.time* prompt respectively¹². In the **1-pass concept** experimental condition (3) we used the *concept-place*,

12. As shown in Tables 4 and 5, some, but not all, utterances in a confirmation state contain the corresponding concept.

| Exp # | Num pass | Predict LM method | Build LM method | Overall | Concept utterances | |
|-------|----------|----------------------------|---------------------|-----------|--------------------|----------------|
| | | | | WER | WER | Concept recall |
| 1 | 1-pass | baseline | baseline | 38.49% | 49.12% | 50.75% |
| 2 | 1-pass | confirm-type | confirm-type | 38.83% | 48.96% | 51.36% |
| 3 | 1-pass | confirm-type | concept-type | 46.47% ** | 50.73% * | 52.9% † |
| 4 | 2-pass | DH3_LEX_RAW | concept-type | 38.48% | 47.56% ** | 53.2% † |
| 5 | 2-pass | ASR_DH3_LEX_RAW | concept-type | 38.51% | 47.99% * | 52.7% |
| 6 | 2-pass | CTM_ASR_DH3_LEX_RAW | concept-type | 38.42% | 47.86% * | 52.6% |
| 7 | 2-pass | oracle | concept-type | 37.85% ** | 45.94% ** | 54.91% ** |

Table 18: Speech recognition results. ** indicates a statistically significant difference ($p < .01$). * indicates a statistically significant difference ($p < .05$). † indicates a near-significant trend in difference ($p < .1$). Significance for WER is computed using paired t-tests. Significance for concept recall is computed as an inference on proportions.

concept-bus and *concept-time* language models to recognize post-confirmation utterances produced following a *confirm_place*, *confirm_bus* and *confirm_time* prompt respectively.

In experimental conditions 4 - 7 we used the 2-pass recognition method outlined in Figure 5. We performed first-pass recognition of post-confirmation utterances using the generic *confirm* language model. Then, we ran the output of the first pass through a concept type classifier. Finally, we performed second-pass recognition using the *concept-place*, *concept-bus* or *concept-time* language models if the utterance was classified as *place*, *bus* or *time* respectively¹³. We experimented with the three classification models with highest overall *rec+* when trained on recognized speech: DH3_LEX_RAW (4), ASR_DH3_LEX_RAW (5), and CTM_ASR_DH3_LEX_RAW (6). To get an idea of “best possible” performance, we also report 2-pass oracle (7) recognition results, assuming an oracle classifier that always outputs the correct concept type for an utterance.

7.2 Experimental Results

7.2.1 COMPARISON OF MODELS

In Table 18 we report average per-utterance word error rate (WER) on post-confirmation utterances, average per-utterance WER on post-confirmation utterances containing a concept, and average concept recall rate (percentage of correctly recognized concepts) on post-confirmation utterances containing a concept. In slot-filling dialogue systems like *Let’s Go!*, the concept recall rate largely determines the potential of the system to understand user-provided information and continue the dialogue successfully. Therefore, our goal is to maximize concept recall and minimize WER on concept-containing utterances, without causing overall WER to decline.

13. We treated utterances classified as containing more than concept type as *none*. In the 2006 data, only 5.6% of utterances with a concept contain more than one concept type.

| | Transcript | Generic model hypothesis | Concept-specific model hypothesis |
|---|------------------------------------|--------------------------|-----------------------------------|
| 1 | NO ARDMORE | NO FIVE MORNING | NO ARDMORE |
| 2 | NO LEAVING FROM HO-BART AND MURRAY | NO LEAVING FROM FOUR A M | NO LEAVING FROM FOR MURRAY |
| 3 | ELEVEN O'CLOCK | D BRADDOCK O'CLOCK | ELEVEN O'CLOCK |
| 4 | FIFTH AND DINWIDDIE | FIFTY THE 1A | FIFTH AND DINWIDDIE |

Table 19: Examples utterances with improved speech recognition performance in the second pass of the speech recognizer for the DH3_LEX_RAW prediction model.

As Table 18 shows, the **1-pass confirm-type** (2) and **1-pass concept-type** (3) experimental recognizers perform better than the baseline recognizer (1) in terms of concept recall, but worse in terms of overall WER. Most of these differences are not statistically significant. However, the **1-pass concept-type** recognizer (3) has significantly worse overall and concept utterance WER than the **baseline** recognizer ($p < .01$). The **1-pass concept-type** recognizer would in practice follow the performance of the *confirm-type* prediction method, which has the highest rate of *switch+* (17%) and *switch* (72%) errors (see Table 11). This is because with the *confirm-type* prediction method all utterances without a concept (82%) are decoded with a language model built on utterances with a concept. The *switch+* error rate for this method indicates that 17% of utterances containing a concept type would be classified as containing a different concept type and decoded with a language model built for that different concept type, leading to poorer speech recognition performance.

All of the 2-pass recognizers (4-7) use automatic concept prediction and achieve significantly lower concept utterance WER than the **baseline** recognizer ($p < .05$). Differences between these recognizers in overall WER and concept recall are not significant. The **2-pass oracle** recognizer (7) shows the best possible improvement from using concept-type language models. It achieves significantly higher concept recall and significantly lower overall and concept utterance WER than the **baseline** recognizer ($p < .01$). It also achieves significantly lower concept utterance WER than any of the 2-pass recognizers that use automatic concept prediction ($p < .01$).

7.2.2 ANALYSIS OF IMPROVEMENTS AND ERRORS

We further analysed the effect of concept type prediction of the 2-pass DH3_LEX_RAW model. Table 19 shows examples of utterances where a correctly predicted concept-specific model improved speech recognition performance. In examples #1 and #2 the user corrects the system by making a negation and specifying a place concept. A generic model incorrectly predicts a time concept in both cases while the concept-specific model correctly recognizes a place concept in #1 and partially recognizes a place concept in #2. Examples #3 and #4 show improvement of speech recognition leading to correct time and place concept detection.

While the overall speech recognition and concept detection rate improves, it is possible for the recognition to degrade when a concept prediction model predicts an incorrect concept. These errors correspond to *switch+* errors in Table 13. The DH3_LEX_RAW prediction model has 7.2% *switch+* errors. Table 20 shows examples of incorrectly recognized utterances following two-stage recognition with the DH3_LEX_RAW prediction model and concept-specific language models. Notice that example #5 shows concept type prediction leading to recognition of a different concept

| | Transcript | Generic model hypothesis | Concept-specific model hypothesis |
|---|-----------------------|--------------------------|-----------------------------------|
| 5 | 1A I NEED THE 1A FROM | 1A ANY ONE LEAVE FROM | WE'RE ME WHAT LEAVE FROM MALL |
| 6 | NO CONWAY | NO MORNING | NO PORT AUTHORITY |
| 7 | EARLY MORNING | TROLLEY MORNING | FROM LEAVE MORNING |

Table 20: Examples utterances with degraded speech recognition performance in the second pass of the speech recognizer for the DH3_LEX_RAW prediction model.

type (bus number *1a* vs. a *place*), while in examples #6 and #7 neither speech recognition process leads to correct recognition, which is likely to happen if an utterance is not spoken clearly or contains background noise. In future work, we may attempt to further reduce the *switch+* error rate by considering multiple speech recognition hypotheses from first-pass and second-pass recognition in parallel.

7.2.3 SUMMARY

Our results with two-pass recognition show that it is possible to use knowledge of (or predictions about) the concepts in a user's utterance to improve speech recognition. Our results with the one-pass concept-type recognizer condition show that this cannot be effectively done by assuming that the user will always address the system's question; instead, one must consider the user's actual utterance and the discourse history (as in the DH3_LEX_RAW model).

8. Discussion and Future Work

In this set of experiments, we looked at user responses to system confirmation prompts in task-oriented spoken dialogue. We explained how these post-confirmation utterances are of outsize importance in spoken dialogue, because they may contain unrequested task-relevant concepts that are likely to be misrecognized, leading to cascading errors and reduced user satisfaction. We then examined one type of responsive adaptation: a task-oriented dialogue system adapting its speech recognizer to handle user responses to system confirmation prompts. We showed that by using acoustic, lexical, dialogue state and dialogue history features, we are able to predict the presence of task-relevant concept types in first-pass speech recognition output for post-confirmation utterances with 93% accuracy. We also showed that use of a concept type predictor can lead to improvements in two-pass speech recognition performance in terms of WER and concept recall.

Of course, any possible improvements in speech recognition performance are dependent on (1) the performance of concept type classification; (2) the accuracy of first-pass speech recognition; and (3) the accuracy of second-pass speech recognition. For example, with the general language model, we get a fairly high overall WER of 38.49%. In future work, we will systematically vary the WER of both the first- and second-pass speech recognizers to further explore the interaction between speech recognition performance and concept type classification.

The improvements the two-pass recognizers achieve have quite small local effects (up to 3.18% absolute improvement in WER on utterances containing a concept, and less than 1% on post-confirmation utterances overall) but may have larger impacts on dialogue completion times and task

completion rates, as they reduce the number of cascading recognition errors in the dialogue (Shin et al. 2002). Furthermore, we could use knowledge of the concept type(s) contained in user utterances to improve dialogue management and response planning (Bohus 2007). In future work, we will look at (1) extending the use of concept-type classifiers to utterances following any system prompt; and (2) the impact of these interventions on overall metrics of dialogue success.

Although this work was carried out with a *flexible input* dialogue system where a user can speak longer phrases and sentences, the proposed approach may also be applied to *fixed input* systems where the vocabulary accepted by a system is limited. Even in *fixed input* systems, a user may attempt corrections, clarifications and topic shifts in post-confirmation utterances. To avoid situations of repetitive “*Sorry I did not understand you*” prompts, a system may attempt to predict the concept type(s) present in post-confirmation utterances and adapt its grammar or language model to decrease the chance of errors due to out-of-vocabulary speech.

Our results also have implications for unconstrained open-domain dialogue systems, such as Turing test candidate systems. Although in an unconstrained dialogue, topics and vocabulary may shift dramatically over time, pairs of consecutive utterances are related to one another and a flow can be traced throughout most coherent dialogues (Schegloff and Sacks 1973). This property of communication allows open-domain systems to use dialogue history to adapt their models for the upcoming utterances making recognition and understanding more tractable.

An alternative method for maximizing recognition performance in dialogue systems is to guide the user to use only desired vocabulary and syntax. In separate research, we are looking at *directive adaptation* in spoken dialogue systems (Stent et al. 2006, Stoyanchev and Stent 2009), exploring the potential impact on the user of micro-level design decisions in system prompt construction.

9. Conclusions

Adaptation is an important feature of successful spoken dialogue. Most dialogue systems use either egocentric adaptation (adaptation of system behavior in response to changes in system internal state) or directive adaptation (adaptation of system behavior intended to cause changes in user behavior). In this paper, we looked at responsive adaptation, adaptation in direct response to user behavior. We examined adaptation in a dialogue situation that is particularly obvious and frustrating for users: speech recognition errors in user utterances in response to system confirmation prompts. We showed that responsive adaptation has the potential to reduce the frequency and severity of these types of error, leading to improved dialogue outcomes.

Of course, there are many other ways to apply adaptation in dialogue systems: for example, a system may modify the type of help it provides in response to different types of processing error (Hockey et al. 2003), or may modify the type of feedback it provides in response to user indications of uncertainty (Forbes-Riley and Litman 2009), or may adjust its words and syntactic choices to match those of the user to seem more “natural” (Dubey et al. 2006a). Each type of adaptation on its own may have only a small impact, but together they have the effect of creating dialogue systems that are easier to use.

10. Acknowledgments

This research was done while both authors were at Stony Brook University in Stony Brook, NY, USA. This material is based upon work supported by the National Science Foundation under Grant

No. 0325188. We thank our collaborators on this project, particularly Drs. Susan Brennan and Marie Huffman. We also thank the developers of the *Let's Go!* system, particularly Drs. Maxine Eskenazi and Antoine Raux, for providing access to their system and data.

References

- F. Bechet, G. Riccardi, and D. Hakkani-Tür. Mining spoken dialog corpora for system evaluation and modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.
- J. R. Bellegarda. Statistical language model adaptation: Review and perspectives. *Speech Communication Special Issue on Adaptation Methods for Speech Recognition*, 42:93–108, 2004.
- B. Bigi, Y. Huang, and R. De Mori. Vocabulary and language model adaptation using information retrieval. In *Proceedings of INTERSPEECH*, 2004.
- A. W. Black, S. Burger, B. Langner, G. Parent, and M. Eskenazi. Spoken dialog challenge 2010. In *Proceedings of the Spoken Language Technology Conference*, 2010.
- P. Boersma and D. Weenink. Praat. <http://www.fon.hum.uva.nl/praat/>.
- D. Bohus. *Error awareness and recovery in task-oriented spoken dialog systems*. PhD thesis, Carnegie Mellon University, 2007.
- D. Bohus and A. Rudnicky. LARRI: A language-based maintenance and repair assistant. In *Proceedings of Multi-Modal Dialogue in Mobile Environments*, 2002.
- D. Bohus and A. Rudnicky. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proceedings of EUROSPEECH*, 2003.
- H. Branigan, M. Pickering, and A. Cleland. Syntactic coordination in dialogue. *Cognition*, 75: B13–B25, 2004.
- S. Brennan and H. Clark. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, 22(6):1482–1493, 1996.
- J. Chu-Carroll and J. Nickerson. Evaluating automatic dialogue strategy adaptation for a spoken dialogue system. In *Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2000.
- M. Danieli and E. Gerbino. Metrics for evaluating dialogue strategies in a spoken language system. In *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, 1995.
- A. Dubey, F. Keller, and P. Sturt. Integrating syntactic priming into an incremental probabilistic parser, with an application to psycholinguistic modeling. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2006a.
- A. Dubey, P. Sturt, and F. Keller. Parallelism in coordination as an instance of syntactic priming: evidence from corpus-based modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006b.

- Y. Esteve, F. Bechet, A. Nasr, and R. De Mori. Stochastic finite state automata language model triggered by dialogue states. In *Proceedings of EUROSPEECH*, 2001.
- E. Filisko and S. Seneff. Developing city name acquisition strategies in spoken dialogue systems via user simulation. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue (SIGDIAL)*, 2005.
- K. Forbes-Riley and D. Litman. Adapting to student uncertainty improves tutoring dialogues. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED)*, 2009.
- Y. Fukubayashi, K. Komatani, T. Ogata, and H. Okuno. Dynamic help generation by estimating user’s mental model in spoken dialogue systems. In *Proceedings of INTERSPEECH*, 2006.
- M. Gabsdil and O. Lemon. Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2004.
- S. Garrod and A. Anderson. Saying what you mean in dialogue: A study of conceptual and semantic coordination. *Cognition*, 27(2):181–218, 1987.
- D. Gildea and T. Hofmann. Topic-based language models using EM. In *Proceedings of EUROSPEECH*, 1999.
- A. L. Gorin, G. Riccardi, and J. H. Wright. How may I help you? *Speech Communication*, 23: 113–127, 1997.
- G. Gorrell, I. Lewin, and M. Rayner. Adding intelligent help to mixed initiative spoken dialogue systems. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2002.
- B. Hockey, O. Lemon, E. Campana, L. Hiatt, G. Aist, J. Hieronymus, A. Gruenstein, and J. Dowding. Targeted help for spoken dialogue systems: intelligent feedback improves naive users’ performance. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*, 2003.
- D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky. Pocket-Sphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2006.
- R. Iyer and M. Ostendorf. Modeling long distance dependencies in language: Topic mixtures versus dynamic cache model. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39, 1999.
- D. Jurafsky, C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchman, and N. Morgan. Using a stochastic context-free grammar as a language model for speech recognition. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1995.
- S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. Comparing grammar-based and robust approaches to speech understanding: A case study. In *Proceedings of EUROSPEECH*, 2001.

- T. Kraljic, A.G. Samuel, and S.E. Brennan. First impressions and last resorts: How listeners adjust to speaker variability. *Psychological Science*, 19:332–338, 2008.
- I. Kruijff-Korabayova and O. Kukina. The effect of dialogue system output style variation on users’ evaluation judgments and input style. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, 2008.
- O. Lemon and I. Konstas. User simulations for context-sensitive speech recognition in spoken dialogue systems. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*, 2009.
- D. Litman, J. Hirschberg, and M. Swerts. Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics*, 32:417–438, 2006.
- C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- C. Martins, A. Teixeira, and J. Neto. Dynamic language modeling for European Portuguese. *Computer Speech and Language*, 24:750–773, 2010.
- S. Oviatt, J. Bernard, and G. A. Levow. Linguistic adaptations during spoken and multimodal error resolution. *Language and Speech*, 41:419–442, 1998.
- A. Raux, B. Langner, A. Black, and M. Eskenazi. Let’s Go Public! taking a spoken dialog system to the real world. In *Proceedings of EUROSPEECH*, 2005.
- G. Riccardi and A.L. Gorin. Stochastic language adaptation over time and state in natural spoken dialog systems. *IEEE Transactions on Speech and Audio Processing*, 8(1):3–10, 2000.
- G. Riccardi and D. Hakkani-Tür. Active and unsupervised learning for automatic speech recognition. In *Proceedings of EUROSPEECH*, Geneva, Switzerland, September 2003.
- A. Rudnicky, C. Bennett, A. Black, A. Chotomongcol, K. Lenzo, A. Oh, and R. Singh. Task and domain specific modelling in the Carnegie Mellon Communicator system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.
- E. Schegloff and H. Sacks. Opening up closings. *Semiotica*, 8:289–327, 1973.
- T. Sheeder and J. Balogh. Say it like you mean it: Priming for structure in caller responses to a spoken dialog system. *International Journal of Speech Technology*, 6(2):103–111, 2003.
- J. Shin, S. Narayanan, L. Gerber, A. Kzetzadeh, and D. Byrd. Analysis of user behavior under error conditions in spoken dialogs. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2002.
- R. Smith and S. Gordon. Effects of variable initiative on linguistic behavior in human-computer spoken natural language dialogue. *Computational Linguistics*, 23(1):141–168, 1997.
- S. Stenchikova, B. Mucha, S. Hoffman, and A. Stent. RavenCalendar: A multimodal dialog system for managing a personal calendar. In *Proceedings of the Human Language Technology Conference (HLT)*, 2007.

- A. Stent, S. Stenichikova, and M. Marge. Dialog systems for surveys: The Rate-a-Course system. In *Proceedings of the IEEE/ACL Spoken Language Technology Workshop (SLT)*, 2006.
- S. Stoyanchev. *Impact of responsive and directive adaptation on local dialog processing*. PhD thesis, Stony Brook University, 2009.
- S. Stoyanchev and A. Stent. Lexical and syntactic priming and their impact in deployed spoken dialog systems. In *Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2009.
- S. Stoyanchev, D. Hakkani-Tür, and G. Tur. Name-aware speech recognition for interactive question answering. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2008.
- S. Tomko and R. Rosenfeld. Shaping user input in speech graffiti: a first pass. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006.
- G. Tur. Extending boosting for large scale spoken language understanding. *Machine Learning*, 69(1):55–74, 2007.
- G. Tur and A. Stolcke. Unsupervised language model adaptation for meeting recognition. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2007.
- G. Tur, D. Hakkani-Tür, and R. E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005.
- M. Walker, A. Rudnicky, R. Prasad, J. Aberdeen, E. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, S. Roukos, G. Sanders, S. Seneff, and D. Stallard. Darpa Communicator: Cross-system results for the 2001 evaluation. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2002.
- W. Ward and S. Issar. Recent improvements in the CMU spoken language understanding system. In *Proceedings of the Human Language Technology Conference (HLT)*, 1994a.
- W. Ward and S. Issar. Integrating semantic constraints into the Sphinx-II recognition search. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1994b.
- I. Witten and F. Eibe. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.
- W. Xu and A. Rudnicky. Language modeling for dialog system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.
- S. Young. Detecting misrecognitions and out-of-vocabulary words. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1994.