Predicting Concept Types in User Corrections in Dialog

Svetlana Stoyanchev and **Amanda Stent**
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400, USA
svetlana.stoyanchev@gmail.com, amanda.stent@stonybrook.edu

Abstract

Most dialog systems explicitly confirm user-provided task-relevant concepts. User responses to these system confirmations (e.g. corrections, topic changes) may be misrecognized because they contain unrequested task-related concepts. In this paper, we propose a *concept-specific language model adaptation strategy* where the language model (LM) is adapted to the concept type(s) actually present in the user's post-confirmation utterance. We evaluate concept type classification and LM adaptation for post-confirmation utterances in the *Let's Go!* dialog system. We achieve 93% accuracy on concept type classification using acoustic, lexical and dialog history features. We also show that the use of concept type classification for LM adaptation can lead to improvements in speech recognition performance.

1 Introduction

In most dialog systems, the system explicitly confirms user-provided task-relevant *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, Oakland") or a *topic change* (e.g. "no, leave at 7" or "yes, and go to Oakland"). Each type of utterance has implications for further processing. In particular, corrections and topic changes are likely to contain unrequested task-relevant concepts that are not well represented in the recognizer's post-confirmation language model (LM)[1]. This means that they are likely to be misrecognized, frustrating the user and leading to cascading errors. Correct determination of the content of post-confirmation utterances can lead to improved speech recognition, fewer and shorter sequences of speech recognition errors, and improved dialog system performance.

In this paper, we look at user responses to system confirmation prompts CMU's deployed *Let's Go!* dialog system. We adopt a two-pass recognition architecture (Young, 1994). In the first pass, the input utterance is processed using a general-purpose LM (e.g. specific to the domain, or specific to the dialog state). Recognition may fail on concept words such as "Oakland" or "61C" , but is likely to succeed on closed-class words (e.g. ”yes”, ”no”, ”and”, ”but”, ”leaving”). If the utterance follows a system confirmation prompt, we then use acoustic, lexical and dialog 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 LM. 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.

The rest of this paper is structured as follows: In Section 2 we discuss related work. In Section 3 we describe our data. In Section 4 we present our concept type classification experiment. In Section 5 we present our LM adaptation experiment. In Section 6 we conclude and discuss future work.

[1] The word error rate on post-confirmation *Let's Go!* utterances containing a concept is 10% higher than on utterances without a concept.

Proceedings of EACL 2009 Workshop on Semantic Representation of Spoken Language - SRSL 2009, pages 42–49,
Athens, Greece, 30 March 2009. ©2009 Association for Computational Linguistics

2 Related Work

When a dialog system requests a confirmation, the user's subsequent corrections and topic change utterances are particularly likely to be misrecognized. Considerable research has now been done on the automatic detection of spoken corrections. Linguistic cues to corrections include the number of words in the post-confirmation utterance and the use of marked word order (Krahmer et al., 2001). Prosodic cues include F0 max, RMS max, RMS mean, duration, speech tempo, and percentage of silent frames(Litman et al., 2006; Hirschberg et al., 2004; Levow, 1998). Discourse cues include the removal, repetition, addition or modification of a concept, the system's dialog act type, and information about error rates in the dialog so far (Krahmer et al., 2001; et al., 2002; Litman et al., 2006; Walker et al., 2000). In our experiments, we use most of these features as well as additional lexical features.

We can use knowledge of the type or content of a user utterance to modify system behavior. For example, in this paper we use the concept type(s) in the user's utterance to adapt the recognizer's LM. It is now common practice to adapt the recognizer to the type, context or style of input speech (Bellegarda, 2004). LM adaptation has been used to improve automatic speech recognition performance in automated meeting transcription (Tur and Stolcke, 2007), speech-driven question answering (Stenchikova et al., 2008), broadcast news recognition (Gildea and Hofmann, 1999), and spoken dialog systems (Tur, 2005). LMs in dialog systems can be adapted to the dialog state (e.g. (Riccardi and Gorin, 2000; Esteve et al., 2001)), the topic (Iyer and Ostendorf, 1999; Gildea and Hofmann, 1999), or the speaker (Tur, 2007).

3 Data

In this experiment we use annotated dialog transcripts and speech from the *Let's Go!* system, a telephone-based spoken dialog system that provides information about bus routes in Pittsburgh (Raux et al., 2005). The data we used comes from the first two months of *Let's Go!* system operation in 2005 (2411 dialogs), and one month in 2006 (1430 dialogs). This data has been transcribed, and annotated by hand for concept types.

In order to provide the user with route information, *Let's Go!* must elicit 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 (see Figure 1). In the annotated transcripts, the following *concepts* are labeled: `neighborhood, place, time, hour, minute, time-of-day`, and `bus`. For our experiments we collapsed these concepts into three *concept types*: *time* , *place* and *bus*.

1	Sys	Welcome to the CMU Let's Go bus information system. What can I do for you?
2	User	I need to go from Oakland**:p**
	ASR	*I need to go .from. can't*
3	Sys	Where do you wanna leave from?
4	User	to Waterfront**:p**
	ASR	*told. .me. Waterfront*
5	Sys	Leaving from WATERFRONT . Did I get that right?
6	User	OAKLAND**:p**
	ASR	*OAKLAND*
7	Sys	Leaving from OAKLAND . Did I get that right?

Figure 1: Dialog extract from *Let's Go!* data. User utterances are annotated with concept types (e.g. :p for place)

Let's Go! has five dialog states corresponding to the type of user utterance it expects: *first-query*, *next-query*, *yes-no*, *place* and *time*. Its speech recognizer uses dialog state-specific n-gram LMs trained on user utterances from the 2005 data. We focus on user utterances in response to system confirmation prompts (the *yes-no* state). Table 1 shows statistics about *yes-no* state utterances in *Let's Go!*. Table 2 shows a confusion matrix for confirmation prompt concept type and post-confirmation utterance concept type. This table indicates the potential for misrecognition of post-confirmation utterances. 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.

In our experiments, we used the 2006 data to train concept type classifiers and for testing. We used the 2005 data to build LMs for our speech recognition experiment.

4 Concept Classification

4.1 Method

Our goal is to classify each post-confirmation user utterance by the concept type(s) it contains (*place, time, bus* or *none*) for later language-model adaptation (see Section 5). From the post-confirmation user utterances in the 2006 dataset described in

Event	2005		2006	
	num	%	num	%
Total dialogs	2411		1430	
Total yes-no confirms	9098	100	9028	100
Yes-no confirms with a concept	2194	24	1635	18.1
Dialog State				
Total confirm place utts	5548	61	5347	59.2
Total confirm bus utts	1763	19.4	1589	17.6
Total confirm time utts	1787	19.6	2011	22.3
Concept Type Features				
Yes-no utts with place	1416	15.6	1007	11.2
Yes-no utts with time	296	3.2	305	3.4
Yes-no utts with bus	584	6.4	323	3.6
Lexical Features				
Yes-no utts with 'yes'	4395	48.3	3693	40.9
Yes-no utts with 'no'	2076	22.8	1564	17.3
Yes-no utts with 'I'	203	2.2	129	1.4
Yes-no utts with 'from'	114	1.3	185	2.1
Yes-no utts with 'to'	204	2.2	237	2.6
Acoustic Features				
feature	mean	stdev	mean	stdev
Duration (seconds)	1.341	1.097	1.365	1.242
RMS mean	.037	.033	.055	.049
F0 mean	183.0	60.86	185.7	58.63
F0 max	289.8	148.5	296.9	146.5

Table 1: Statistics on post-confirmation utterances

	place	bus	time
2005 dataset			
confirm_place	0.86	0.13	0.01
confirm_bus	0.18	0.81	0.01
confirm_time	0.07	0.01	0.92
2006 dataset			
confirm_place	0.87	0.10	0.03
confirm_bus	0.34	0.64	0.02
confirm_time	0.15	0.13	0.71

Table 2: Confirmation state vs. user concept type

Section 3, we extracted the features described in Section 4.2 below. To identify the correct concept type(s) for each utterance, we used the human annotations provided with the data.

We performed a series of 10-fold cross-validation experiments to examine the impact of different types of feature on concept type classification. 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 Frank, 2005)[2].

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*)[3]. We also report overall *pre*+, *rec*+, f-measure (*f*+), and classification accuracy across the three concept types. Finally, we report the percentage of *switch*+ errors and *switch* errors. *Switch*+ errors are utterances containing *bus* classified as *time/place*, *time* as *bus/place*, and *place* as *bus/time*; these are the errors most likely to cause decreases in speech recognition accuracy after language model adaptation. *Switch* errors include utterances with no concept classified as *place*, *bus* or *time*.

Only utterances classified as containing one of the three concept types are subject to second-pass recognition using a concept-specific language model. Therefore, these are the only utterances on which speech recognition performance may improve. This means that we want to maximize *rec*+ (proportion of utterances containing a concept that are classified correctly). On the other hand, utterances that are incorrectly classified as containing a particular concept type will be subject to second-pass recognition using a poorly-chosen language model. This may cause speech recognition performance to suffer. This means that we want to minimize *switch*+ errors.

4.2 Features

We used the features summarized in Table 3. All of these features are available at run-time and so may be used in a live system. Below we give additional information about the RAW and LEX features; the other feature sets are self-explanatory.

4.2.1 Acoustic and Dialog History Features

The acoustic/prosodic and dialog history features are adapted from those identified in previous work on detecting speech recognition errors (particularly (Litman et al., 2006)). We anticipated that these features would help us distinguish corrections and rejections from confirmations.

4.2.2 Lexical Features

We used lexical features from the user's current utterance. Words in the output of first-pass ASR are highly indicative both of concept presence or absence, and of the presence of particular concept types; for example, *going to* suggests the presence of a *place*. We selected the most salient lexi-

[2] J48 gave the highest classification accuracy compared to other machine learning algorithms we tried on this data.

[3] We do not report precision or recall for determining *absence* of each concept type. In our data set 82.2% of the utterances do not contain any concepts (see Table 1). Consequently, precision and recall for determining absence of each concept type are above .9 in each of the experiments.

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

Table 3: Features for concept type classifiers

cal features (unigrams and bigrams) for each concept type by computing the *mutual information* between potential features and concept types (Manning et al., 2008). For each lexical feature t and each concept type class $c \in \{$ *place* $+$, *place* -, *time* $+$, *time* -, *bus* $+$, *bus* -$\}$, we computed *I*:

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

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.

To identify the most relevant lexical features, we extracted from the data all the transcribed user utterances. We removed all words that realize concepts (e.g. "61C", "Squirrel Hill"), as these are likely to be misrecognized in a post-confirmation utterance. We then extracted all word unigrams and bigrams. We computed the mutual information between each potential lexical feature and concept type. We then selected the 30 features with the highest mutual information which occurred at least 20 times in the training data[4].

For transcribed speech only, we also compute the string edit distance between the current and previous user utterances. This gives some indication of whether the current utterance is a correction or topic change (vs. a confirmation). However, for recognized speech recognition errors reduce the effectiveness of this feature (and of the concept features in the dialog history feature set).

4.3 Baseline

A simple baseline for this task, **No-Concept**, always predicts *none* in post-confirmation utterances. This baseline achieves overall classification accuracy of 82% but *rec+* of 0. At the other extreme, the **Confirmation State** baseline assigns to each utterance the dialog system's confirmation prompt type (using the DIA feature). This baseline achieves *rec+* of .79, but overall classification accuracy of only 14%. In all of the models used in our experiments, we include the current confirmation prompt type (DIA) feature.

4.4 Experiment Results

In this section we report the results of experiments on concept type classification in which we examine the impact of the feature sets presented in Table 3. We report performance separately for recognized speech, which is available at runtime (Table 5); and for transcribed speech, which gives us an idea of best possible performance (Table 4).

4.4.1 Features from the Current Utterance

We first look at lexical (LEX) and prosodic (RAW) features from the current utterance. For both recognized and transcribed speech, the LEX model achieves significantly higher *rec+* and overall accuracy than the RAW model ($p < .001$). For recognized speech, however, the LEX model has significantly more *switch+* errors than the RAW model ($p < .001$). This is not surprising since the majority of errors made by the RAW model are labeling an utterance with a concept as *none*. Utterances misclassified in this way are not subject to second-pass recognition and do not increase WER.

[4] We aimed to select equal number of features for each class with information measure in the top 25%. 30 was an empirically derived threshold for the number of lexical features to satisfy the desired condition.

Features	Place			Time			Bus			Overall					
	pre+	rec+	acc	pre+	rec+	acc	pre+	rec+	acc	pre+	rec+	f+	acc	switch+	switch
No Concept	0	0	.86	0	0	0.81	0	0	.92	0	0	0	**0.82**	0	0
Confirmation State	0.87	0.85	0.86	0.64	0.54	0.58	0.71	0.87	0.78	**0.14**	**0.79**	**0.24**	0.14	17	72.3
RAW	0.65	0.53	0.92	0.25	0.01	0.96	0.38	0.07	0.96	0.67	0.34	0.45	0.85	**6.43**	4.03
LEX	0.81	0.88	0.96	0.77	0.48	0.98	0.83	0.59	0.98	0.87	**0.72**	**0.79**	**0.93**	7.32	3.22
LEX_RAW	0.83	0.84	0.96	0.75	0.54	0.98	0.76	0.59	0.98	**0.88**	0.70	0.78	**0.93**	7.39	**3.00**
DH1_LEX	0.85	0.91	0.97	0.72	0.63	0.98	0.89	0.83	0.99	0.88	**0.81**	**0.84**	**0.95**	5.48	2.85
DH3_LEX	0.85	0.87	0.97	0.72	0.59	0.98	0.92	0.82	0.99	**0.89**	0.78	0.83	0.94	**5.22**	**2.62**

Table 4: 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			Overall					
	pre+	rec+	acc	pre+	rec+	acc	pre+	rec+	acc	pre+	rec+	f+	acc	switch+	switch
No Concept	0	0	.86	0	0	0.81	0	0	.92	0	0	0	**0.82**	**0**	**0**
Confirmation State	0.87	0.85	0.86	0.64	0.54	0.58	0.71	0.87	0.78	**0.14**	**0.79**	**0.24**	0.14	17	72.3
RAW	0.65	0.53	0.92	0.25	0.01	0.96	0.38	0.07	0.96	0.67	0.34	0.45	0.85	**6.43**	**4.03**
LEX	0.70	0.70	0.93	0.67	0.15	0.97	0.65	0.62	0.98	0.75	0.56	0.64	0.89	9.94	4.93
LEX_RAW	0.70	0.72	0.93	0.66	0.38	0.97	0.68	0.57	0.98	**0.76**	**0.60**	**0.67**	**0.90**	10.32	5.10
DH1_LEX_RAW	0.71	0.68	0.93	0.68	0.38	0.97	0.78	0.63	0.98	**0.77**	0.60	0.67	**0.90**	8.15	**4.55**
DH3_LEX_RAW	0.71	0.70	0.93	0.67	0.42	0.97	0.79	0.63	0.98	**0.77**	**0.62**	**0.68**	**0.90**	**7.20**	4.57
ASR_DH3_LEX _RAW	0.71	0.70	0.93	0.69	0.42	0.97	0.79	0.63	0.98	**0.77**	**0.62**	**0.68**	**0.90**	**7.20**	**4.54**
CTM_DH3_LEX _RAW	0.82	0.82	0.96	0.86	0.71	0.99	0.76	0.68	0.98	**0.85**	**0.74**	**0.79**	**0.93**	**3.89**	**2.94**
CTM_ASR_DH3 _LEX_RAW	0.82	0.81	0.96	0.86	0.69	0.99	0.76	0.68	0.98	**0.85**	**0.74**	**0.79**	**0.93**	4.27	3.01

Table 5: Concept type classification results: recognized speech (all models include feature DIA). Best overall values in each group are highlighted in bold.

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+* and overall accuracy than LEX ($p < .001$). Lexical content from transcribed speech is a very good indicator of concept type. However, lexical content from recognized speech is noisy, so concept type classification from ASR output can be improved by using acoustic/prosodic features.

We note that 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 better overall accuracy and fewer *switch+* errors ($p < .001$) .

4.4.2 Features from the Dialog History

Next, we add features from the dialog history to our best-performing models so far. For transcribed speech, DH1_LEX performs significantly better than LEX in terms of overall accuracy, *rec+*, and *switch+* errors ($p < .001$). DH3_LEX performs significantly worse than DH1_LEX in terms of *rec+* ($p < 0.05$). For recognized speech, 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 ($p < .05$). There are no significant performance differences between DH1_LEX_RAW and DH3_LEX_RAW.

4.4.3 Features Specific to Recognized Speech

Finally, we add the ASR and 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.

We hypothesized that the CTM feature will improve cases where a part of (but not the whole) concept instance is recognized in first-pass recognition[5]. The generic language model used in first-pass recognition recognizes some concept-related words. So, if in the utterance *Madison avenue*, *avenue* (but not *Madison*), is recognized in the first-pass recognition, the CTM feature can flag the utterance with a partial match for *place*, helping the classifier to correctly assign the *place*

[5] We do not try the CTM feature on transcribed speech because there is a one-to-one correspondence between presence of the concept and the CTM feature, so it perfectly indicates presence of a concept.

type to the utterance. Then, in the second-pass recognition the utterance will be decoded with a *place* concept-specific language model, potentially improving speech recognition performance. Adding the CTM feature to DH3_LEX_RAW and ASR_DH3_LEX_RAW leads to a large statistically significant improvement in all measures: a 12% absolute increase in *rec+*, a 3% absolute increase in overall accuracy, and decreases in *switch+* errors ($p < .001$). There are no statistically significant differences between these two models.

4.4.4 Summary and Discussion

In this section we evaluated different models for concept type classification. The best performing transcribed speech model, DH1_LEX, significantly outperforms the **Confirmation State** 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 **Confirmation State** 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 ($p < .01$).

5 Speech Recognition Experiment

In this section we report the impact of concept type prediction on recognition of post-confirmation utterances in *Let's Go!* system data. We hypothesized that speech recognition performance for utterances containing a concept can be improved with the use of concept-specific LMs. We (1) compare the existing *dialog state-specific* LM adaptation approach used in *Let's Go!* with our proposed *concept-specific* adaptation; (2) compare two approaches to *concept-specific* adaptation (using the system's confirmation prompt type and using our concept type classifiers); and (3) evaluate the impact of different concept type classifiers on *concept-specific* LM adaptation.

5.1 Method

We used the PocketSphinx speech recognition engine (et al., 2006) with gender-specific telephone-quality acoustic models built for Communicator (et al., 2000). We trained trigram LMs using 0.5 ratio discounting with the CMU language modeling toolkit (Xu and Rudnicky, 2000)[6]. We built state- and concept-specific hierarchical LMs from the *Let's Go!* 2005 data. The LMs are built with *[place], [time]* and *[bus]* submodels.

We evaluate speech recognition performance on the post-confirmation user utterances from the 2006 testing dataset. Each experiment varies in 1) the LM used for the final recognition pass and 2) the method of selecting a LM for use in decoding.

5.1.1 Language models

We built seven LMs for these experiments. The *state-specific* LM contains all utterances in the training data that were produced in the *yes-no* dialog state. The *confirm-place*, *confirm-bus* and *confirm-time* LMs contain all utterances produced in the *yes-no* dialog state following *confirm_place*, *confirm_bus* and *confirm_time* system confirmation prompts respectively. Finally, the *concept-place*, *concept-bus* and *concept-time* LMs contain all utterances produced in the *yes-no* dialog state that contain a mention of a *place, bus* or *time*.

5.1.2 Decoders

In the baseline, **1-pass general** condition, we use the *state-specific* LM to recognize all post-confirmation utterances. In the **1-pass state** experimental condition we use the *confirm-place, confirm-bus* and *confirm-time* LMs to recognize testing utterances produced following a *confirm_place*, *confirm_bus* and *confirm_time* prompt respectively[7]. In the **1-pass concept** experimental condition we use the *concept-place, concept-bus* and *concept-time* LMs to recognize testing utterances produced following a *confirm_place*, *confirm_bus* and *confirm_time* prompt respectively.

In the *2-pass* conditions we perform first-pass recognition using the *general* LM. Then, we classify the output of the first pass using a concept type classifier. Finally, we perform second-pass recognition using the *concept-place, concept-bus* or *concept-time* LMs if the utterance was classified as *place, bus* or *time* respectively[8]. We used the three classification models with highest overall *rec+*: DH3_LEX_RAW, ASR_DH3_LEX_RAW,

[6] We chose the same speech recognizer, acoustic models, language modeling toolkit, and LM building parameters that are used in the live *Let's Go!* system (Raux et al., 2005).

[7] As we showed in Table 2, most, but not all, utterances in a confirmation state contain the corresponding concept.

[8] We treat 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.

Recognizer	Concept type classifier	Language model	Overall	Concept utterances	
			WER	WER	Concept recall
1-pass	general	state-specific	38.49%	49.12%	50.75%
1-pass	confirm state	confirm-{place,bus,time}	38.83%	48.96%	51.36%
1-pass	confirm state	concept-{place,bus,time}, state-specific	46.47% ♠	50.73% ♣	52.9% ∗
2-pass	**DH3_LEX_RAW**	concept-{place,bus,time}, state-specific	38.48%	47.56% ♠	53.2% ∗
2-pass	**ASR_DH3_LEX_RAW**	concept-{place,bus,time}, state-specific	38.51%	47.99% ♣	52.7%
2-pass	**CTM_ASR_DH3_LEX_RAW**	concept-{place,bus,time}, state-specific	38.42%	47.86% ♣	52.6%
2-pass	oracle	concept-{place,bus,time}, state-specific	37.85% ♠	45.94% ♠	54.91% ♠

Table 6: Speech recognition results. ♠ indicates significant difference (p<.01). ♣ indicates significant difference (p<.05). * indicates near-significant trend in difference (p<.07). Significance for WER is computed as a paired t-test. Significance for concept recall is an inference on proportion.

and CTM_ASR_DH3_LEX_RAW. To get an idea of "best possible" performance, we also report 2-pass oracle recognition results, assuming an oracle classifier that always outputs the correct concept type for an utterance.

5.2 Results

In Table 6 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 dialog systems like *Let's Go!*, the concept recall rate largely determines the potential of the system to understand user-provided information and continue the dialog successfully. Our goal is to maximize concept recall and minimize concept utterance WER, without causing overall WER to decline.

As Table 6 shows, the **1-pass state** and **1-pass concept** recognizers perform better than the **1-pass general** recognizer 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** recognizer has significantly worse overall and concept utterance WER than the **1-pass general** recognizer ($p < .01$).

All of the 2-pass recognizers that use automatic concept prediction achieve significantly lower concept utterance WER than the **1-pass general** recognizer ($p < .05$). Differences between these recognizers in overall WER and concept recall are not significant.

The **2-pass oracle** recognizer achieves significantly higher concept recall and significantly lower overall and concept utterance WER than the **1-pass general** recognizer ($p < .01$). It also achieves significantly lower concept utterance WER than any of the 2-pass recognizers that use automatic concept prediction ($p < .01$).

Our **2-pass concept** results show that it is possible to use knowledge of the concepts in a user's utterance to improve speech recognition. Our **1-pass concept** results 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 our DH3_LEX_RAW model).

6 Conclusions and Future Work

In this paper, we examined user responses to system confirmation prompts in task-oriented spoken dialog. We showed that these post-confirmation utterances may contain unrequested task-relevant concepts that are likely to be misrecognized. Using acoustic, lexical, dialog state and dialog history features, we were able to classify task-relevant concepts in the ASR output for post-confirmation utterances with 90% accuracy. We showed that use of a concept type classifier can lead to improvements in 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 the first-pass speech recognition; and (3) the accuracy of the second-pass speech recognition. For example, with our 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 our 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 impact on dialog completion times and task completion rates, as they reduce the number of cascading recognition errors in the dialog (et al., 2002). Furthermore, we could also use knowledge of the concept type(s) contained in a user utterance to improve dialog management and response planning (Bohus, 2007). In future work, we will look at (1) extending the use of our concept-type classifiers to utterances following any system prompt; and (2) the impact of these interventions on overall metrics of dialog success.

7 Acknowledgements

We would like to thank the researchers at CMU for providing the *Let's Go!* data and additional resources.

References

J. R. Bellegarda. 2004. Statistical language model adaptation: Review and perspectives. *Speech Communication Special Issue on Adaptation Methods for Speech Recognition*, 42:93–108.

D. Bohus. 2007. *Error awareness and recovery in task-oriented spoken dialog systems*. Ph.D. thesis, Carnegie Mellon University.

Y. Esteve, F. Bechet, A. Nasr, and R. Mori. 2001. Stochastic finite state automata language model triggered by dialogue states. In *Proceedings of Eurospeech*.

A. Rudnicky et al. 2000. Task and domain specific modelling in the Carnegie Mellon Communicator system. In *Proceedings of ICSLP*.

J. Shin et al. 2002. Analysis of user behavior under error conditions in spoken dialogs. In *Proceedings of ICSLP*.

D. Huggins-Daines et al. 2006. Sphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of ICASSP*.

D. Gildea and T. Hofmann. 1999. Topic-based language models using EM. In *Proceedings of Eurospeech*.

J. Hirschberg, D. Litman, and M. Swerts. 2004. Prosodic and other cues to speech recognition failures. *Speech Communication*, 43:155–175.

R. Iyer and M. Ostendorf. 1999. Modeling long distance dependencies in language: Topic mixtures versus dynamic cache model. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39.

E. Krahmer, M. Swerts, M. Theune, and M. Weegels. 2001. Error detection in spoken human-machine interaction. *International Journal of Speech Technology*, 4(1).

G.-A. Levow. 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. In *Proceedings of COLING-ACL*.

D. Litman, J.Hirschberg, and M. Swerts. 2006. Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics*, 32:417–438.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

A. Raux, B. Langner, A. Black, and M Eskenazi. 2005. Let's Go Public! Taking a spoken dialog system to the real world. In *Proceedings of Eurospeech*.

G. Riccardi and A. L. Gorin. 2000. Stochastic language adaptation over time and state in a natural spoken dialog system. *IEEE Transactions on Speech and Audio Processing*, 8(1):3–9.

S. Stenchikova, D. Hakkani-Tür, and G. Tur. 2008. Name-aware speech recognition for interactive question answering. In *Proceedings of ICASSP*.

G. Tur and A. Stolcke. 2007. Unsupervised language model adaptation for meeting recognition. In *Proceedings of ICASSP*.

G. Tur. 2005. Model adaptation for spoken language understanding. In *Proceedings of ICASSP*.

G. Tur. 2007. Extending boosting for large scale spoken language understanding. *Machine Learning*, 69(1):55–74.

M. Walker, J. Wright, and I. Langkilde. 2000. Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system. In *Proceedings of ICML*.

I. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.

W. Xu and A. Rudnicky. 2000. Language modeling for dialog system. In *Proceedings of ICSLP*.

S. Young. 1994. Detecting misrecognitions and out-of-vocabulary words. In *Proceedings of ICASSP*.