Impact of Responsive and Directive Adaptation on Local Dialog Processing

A Dissertation Presented

by

Svetlana Stoyanchev

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2009

Stony Brook University

The Graduate School

Svetlana Stoyanchev

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Dr. Amanda Stent - Dissertation Advisor
Computer Science Department

Professor David Warren - Chairperson of Defense
Computer Science Department

Professor Susan Brennan
Psychology Department

Professor Julia Hirschberg
Columbia University
Computer Science Department

This dissertation is accepted by the Graduate School

Lawrence Martin

Dean of the Graduate School

Abstract of the Dissertation

Impact of Responsive and Directive Adaptation on Local Dialog Processing

by

Svetlana Stoyanchev

Doctor of Philosophy

in

Computer Science

Stony Brook University

2009

Spoken dialog systems allow users to access information and accomplish tasks using speech. Understanding and interpreting complex and ambiguous natural language phrases is a challenging task for these systems. Adaptation (the phenomenon of one conversational partner's behavior causing changes in the behavior of the other conversational partner) can be a powerful tool to improve dialog system performance. In this work I examine communication errors in human-computer dialog. I explore the role of directive adaptation (in which the dialog system's behavior guides the user's behavior) and responsive adaptation (in which the user's behavior affects the system's behavior) in avoiding and fixing these errors. The goal of this work is not to model human interaction, but to design methods for improving dialog systems informed by the model of human communication.

The contributions of this thesis include 1) a computational analysis of adaptation

in dialog, 2) experiments evaluating user adaptation to the form of system prompts, and 3) experiments evaluating the effect of system adaptation to the content of user utterances.

In the first study, I compare two possible explanations for adaptation in dialog: *partner design* and *recency*. I propose a new measure of adaptation and use it in a study of the Communicator human-computer spoken dialog corpus to compare strength of adaptation due to *recency* and to *partner design*.

In the second set of studies, I examine user adaptation to the system's lexical and syntactic choices in the context of the deployed *Let's Go!* dialog system. I show that in deployed dialog systems with real users, as in laboratory experiments, users adapt to the system's lexical and syntactic choices. I also show that system prompt formulation can be used to guide users into producing utterances conducive to task success.

In the third set of studies, I evaluate the effect on speech recognition performance of language model adaptation to the task-related topic and content of user utterances. I show that lexical and dialog history features are useful in prediction of utterance content and that the prior knowledge of the content of a user utterance can lead to improvements in speech recognition performance.

Contents

List of Tables

List of Figures

Acknowledgements

First of all I would like to thank my advisor Amanda Stent for helping me in every aspect of the PhD program. She gave me support, inspiration, valuable advice and guidance, as well as the freedom to explore different research directions during my studies. She has helped to broaden my views in research, and inspired me to pursue exciting research ideas. I appreciate and value every advice that she has given me. I am grateful to her being available and helpful at critical moments, and for reading and editing all of my writing. Besides being a wonderful academic advisor, Amanda helped me to gain professional confidence and achieve everything that I have achieved so far.

I am grateful to my family who have always encouraged me to pursue my path and provided me with emotional support. My father Georgi is always happy to give me academia-related advice, to have long conversations about research directions, and discussions about experimental results. He has often listened and gave feedback on my practice talks. My mother Tania reminded me that there is more to life than Ph.D. Special thanks to my loving and supportive husband Rossen who never stopped believing in me and played a key role in steering my career towards research. He has often helped me with editing, cooked many wonderful meals, and shared household work while enduring my repetitive conversations with automatic dialog systems.

I would like to thank the members of my committee Susan Brennan, David Warren, and Julia Hirschberg for generously agreeing to be on my committee and giving me valuable feedback and comments. I am grateful to Susan for providing a psycholinguistic perspective to my research; for giving me an opportunity to participate in the seminars at the psychology department that helped shape my understanding of the field.

Thanks to all my colleagues at the Stony Brook Computer Science and in Psychology departments, especially Anna Kuhlen and Alexia Galati for exciting interdisciplinary discussions.

Special thanks to Carnegie Melon Speech group who provided the Lets Go system and corpora making the experiments presented here possible. Special thanks to Dan Bohus, Antoine Raux, Brian Langner, Maxine Eskenazi for helping me with using Lets Go! system and for giving me interesting ideas and feedback about my experiments.

I had an opportunity to do two internships that gave me a valuable perspective on research in the industry. I am thankful to my wonderful collaborators Gokhan Tur, Dilek Tur, and Jana Sukkarieh with whom I have continued communicating and collaborating after the end of the internship.

Curriculum Vitae

EDUCATION

Stony Brook University, New York — Sep2004 – present
PhD, Computer Science

New York University, New York — Sep 1999 - Jan 2003
MS, Computer Science (part-time study)

University of Maryland, College Park — Sep 1993 - May 1997
BS, Computer Science

RESEARCH EXPERIENCE

State University of New York, Stony Brook — 2004 - present
Computer Science Department, Research Assistant
Advisor: Amanda Stent

Educational Testing Services — Summer 2008
Summer Intern, Mentor: Jana Sukkarieh

AT&T Research Labs — Summer 2005

Summer Intern, Mentors: Gokhan Tur, Dilek Hakkani-Tür

New York University June 2002 - May 2004

Computer Science Department Research Assistant

Advisor: Dan Melamed

TEACHING EXPERIENCE

State University of New York, Stony Brook Spring 2004 / 2005

Teaching Assistant

PROFESSIONAL EXPERIENCE

Online Retail Partners, Java developer Jun 2000 - Apr 2001

Mobius Management Systems, C++ developer Nov 1998 - May 2000

In:sync Corporation, C/C++ developer Jan 1996 - Nov 1998

PUBLICATIONS

- Lexical and Syntactic Priming and Their Impact in Deployed Spoken Dialog Systems *S. Stoyanchev and A. Stent* In Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies, 2009

- Predicting Concept Types in User Corrections in Dialog *S. Stoyanchev and A. Stent* In Proceedings of the EACL Workshop on Semantic Representation of Spoken Language, 2009

- Exact Phrases in Information Retrieval for Question Answering *S. Stoyanchev,* *Y. C. Song, and W. Lahti* In Proceedings of the Coling Workshop on Information Retrieval for Question Answering, 2008

- Name-Aware Speech Recognition for Interactive Question Answering *S. Stoyanchev(Stenchikova), D. Hakkani-Tur, and G. Tur* In Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP), 2008

- Measuring Adaptation Between Dialogs. *S. Stenchikova and A. Stent* In Proceedings of the 7th SIGdial Workshop on Discourse and Dialog, 2007

- RavenCalendar: A Multimodal Dialog System for Managing a Personal Calendar. *S. Stenchikova, B. Mucha, S. Hoffman, and A. Stent* In proceedings of the NAACL HLT Demonstration Program, 2007

- Dialog Systems for Surveys: the Rate-a-Course System *A. Stent, S. Stenchikova, and M. Marge* In Proceedings of the 1st IEEE/ACL Workshop on Spoken Language Technology. SLT, 2006.

- QASR: Question Answering Using Semantic Roles for Speech Interface *S. Stenchikova, D. Hakkani-Tur, and G. Tur* in Proceedings of the ICSLP-Interspeech, September 2006

- QASR: Spoken Answering Using Semantic Role Labeling. *S. Stenchikova, D. Hakkani-Tur, G. Tur ASRU-2005* In proceedings of the 9th biannual IEEE workshop on Automatic Speech Recognition and Understanding, 2005

PROFESSIONAL ACTIVITIES

- Student co-chair of the 2009 NAACL/HLT Student Research Workshop

- Staff Reporter of the IEEE Speech and Language Technical Committee e-Newsletter (2006 - present)

- Organizer of the Graduate Student Research Conference at SUNY, Stony Brook (Spring 2007)

- Organizer of the weekly Stony Brook Natural Language Processing colloquia (Fall 2006, Fall 2007)

- Reviewer for ACL2007 and ICASSP 2007

- Conference Volunteer: HLT/EMNLP 2005, HLT/NAACL 2006

Chapter 1

Introduction

Numerous psycholinguistic studies have demonstrated that people adapt their language use in conversation to that of their conversational partners. For example, user studies have shown that conversational partners adapt to each other's choice of words, particularly referring expressions (Brennan and Clark, 1996), converge on certain syntactic choices (Pickering et al., 2000), adapt to conversation partner's needs Lockridge and Brennan (2002), adapt their prosody to help their partners disambiguate syntactic ambiguities (Kraljic and Brennan, 2005), and also adapt using audiovisual information (Kraut et al., 2003). Some of these results have been duplicated using corpus studies: researchers have found evidence of within-speaker and between-speaker convergence to certain syntactic constructions in dialog corpora (Dubey et al., 2006b; Reitter et al., 2006a). Finally, there is some evidence that people adapt their language use in conversation with computer partners. For example, researchers have shown that users of dialog systems adapt to the system's choice of referring expressions (Brennan, 1991, 1996), the system's choice of modality for referring (Bell et al., 2000; Skantze, 2002), and the system's choice of words (Gustafson et al., 1997).

Spoken dialog systems allow users to access computer interfaces using speech. The richness of natural language allows for great variability in the grammar and vocabulary of user utterances to dialog systems. In conversation with a **flexible input** dialog system (a system that allows user to say phrases and full sentences), a user is not restricted to following a predefined grammar. For example, in a bus information system the response to the prompt "How can I help you" can be a general specification of the task, e. g. *"scheduling"*, a general request for information on a particular bus route, e.g. *"i need /uh/ information on the 56e"*, or a question about a specific route, e.g. *"when is the next 28x from downtown to the airport?"*. Variability is possible even in simple utterances specifying a single **concept** such as time: *four pm*, *four o'clock*, *around four*. This variability in the use of natural language complicates the dialog system designer's task and often causes misrecognitions.

In this thesis I explore the possibility of using adaptation to improve the user experience in dialog with automatic dialog systems. I address the question of adaptation from the three different perspectives. First, I study adaptation in a human-human dialog corpus. Second, I perform an empirical study of the user adaptation to system prompts and the effect of system prompts on the local dialog processing. Third, I evaluate the effect of system adaptation to user utterances on system performance.

Adaptation in dialog is exhibited by the **convergence** of language use and interactive behavior of two agents in dialog. This convergence can be lexical, syntactic, semantic, prosodic, or acoustic. In my work I focus on studying lexical and syntactic convergence in dialog. Convergence is possible when there is variability. We can observe convergence (or lack of it) when a speaker has an option of choosing among alternative expressions, such as synonymous words, tenses, or concepts. A speaker converges to the conversation partner if the speaker makes a linguistic choice by following the partner.

Psycholinguistic researchers hypothesize that speakers maintain **a model** that affects their linguistic choices. Studies of adaptation in the psycholinguistics literature identify two potential contributions to convergence: dialog **partner** and **recency**.

A **partner model** is structured information pertinent to the current dialog and held by a dialog participant about his/her dialog partner. A partner model may store partner-specific information pertaining to lexical, syntactic, semantic, or prosodic preferences. For example, a model may record how the partner refers to a particular object, which tense the partner prefers to use, the prosodic characteristics of the partner's utterances, or the native language of the partner. This model may be partially constructed from prior knowledge about the dialog partner, e.g. from previous conversations. The model is dynamically updated in the course of a dialog. Researchers hypothesize that partners in human-human dialogs maintain models of their dialog partners. In human-computer dialog the user maintains a model of the system (the **system model**) and the system maintains a model of each user or group of users (the **user model**). The system consults the **user model** to customize its behavior to the user, for example, generating responses tailored to a specific user, or adjusting automatic speech recognizer to fit a specific user action. In the psycholinguistic literature, adaptation due to a partner model is referred to as the **partner effect** or **partner-specific adaptation**.

A **recency model** stores information pertinent to the most recent utterance(s) in a dialog. A recency model may store the same information as the partner model, but it does not maintain separate models for each partner. This model is constructed from the most recent utterances in the dialog. In the psycholinguistic literature, adaptation to the most recent utterance(s) in a dialog is refereed to as the **recency effect**, **recency-specific adaptation**, or **convergence**.

The term **priming** is predominantly used in the context of adaptation due to recency in the psycholinguistic literature. It refers to the behavior that is later adapted/repeated (behavior is faster/easier when repeated or after exposure to a stimulus).

In this thesis I consider both partner and recency adaptation. I use the term **priming** to refer to any event in a dialog that influences linguistic decision-making. An instance of priming occurs when a syntactic structure or lexical item giving evidence of a linguistic choice (prime) influences the recipient to make the same decision, i.e. re-use the structure, at a later choice-point (target) (Reitter et al., 2006b). I use the term **convergence** to refer to the effect of either partner or recency adaptation. **Convergence** occurs when dialog participants change their language use to be more similar to each others over time. I use the term **prime** to indicate an utterance, a dialog, or a document that contains priming features. I use the term **target** to indicate an utterance, a dialog, or a document in which convergence for these features is measured. I use the term **adaptation** to indicate the direction of convergence. So, for example, the prime is in the system utterance and the target is a user utterance, I look for user adaptation to the system.

My goal is to evaluate the effect of lexical and syntactic **priming**. I analyze user adaptation to the system's choice of verbs, prepositions, and the form of task-related **concepts**. A **concept** in this context is a piece of information that a system has to obtain from a user. In an flight booking system concepts would include departure city, date, time, airline, etc. Some **concepts** can be specified in a number of ways (e.g. *four pm/four o'clock/four*). Variability in the forms specifying a concept allows for adaptation in user utterances. I also look at system adaptation to the user's choices of form for task-related concepts. In this thesis I do not presume that either **partner** or **recency** adaptation is the only cause of adaptation in dialog. After all, a computer partner can build partner and recency models simultaneously. Instead, I look for convergence in dialog, and examine the impact of computational modeling of convergence on dialog system performance.

My research objective is to understand convergence in human-computer dialogs and its utility for the human-computer communication. I focus on two types of adaptation in system behavior: **responsive adaptation**, in which the system modifies its own behavior to better fit the user's behavior; and **directive adaptation**, in which the system uses behaviors designed to guide the user into less error-prone behaviors. **Directive adaptation** aims at directing the user explicitly or implicitly into adapting to the system or using vocabulary and syntax that may be more easily processed by the system. Directive adaptation may be accomplished through **directive prompts**, system prompts that use a particular form of a concept or specific words to **prime** the user and direct them into specific syntax or vocabulary. Every prompt has some **directive power**, or ability to guide the user. In my work I evaluate the **directive power** of system prompts. **Responsive adaptation** in a dialog system involves adjusting system components to a particular user or a dialog situation. This may involve language or acoustic model adaptation in the speech recognition component, or adaptation of a dialog policy in the dialog manager. In my work I evaluate speech recognition improvement by adapting the dialog system's language model and increasing the context sensitivity of the recognizer.

1.1 Thesis Contributions

The main contributions of this work are:

- A study measuring partner and recency adaptation in human-human dialog.
    - I designed two new measures of adaptation between dialogs. The new measures take into account frequency of a feature in the prime and target documents (while the previous measures took into account only presence of a feature).
    - I compared recency and partner-model adaptation. I found that speakers

exhibit adaption to both types. I outlined differences in the features that are adapted to recency and to the partner model: syntactic features tend to be adapted to the most recent partner. Lexical features with a personal pronoun '*I*' are adapted to the most recent partner. Features indicating direction (*across, through, about the*), and features with the pronoun '*you*' tend to be adapted to a specific partner, regardless of recency.

- An empirical study of user adaptation to system prompts using a live spoken dialog system and real users. I examined how users can be guided into using specific words, syntax, and concept forms.
  - I collected and transcribed a spoken dialog corpus using CMU's deployed *Let's Go!* system.
  - I found that users adapt to 1) verbs and prepositions in the system prompt; and 2) function verb form in the system prompt.
  - I measured adaptation to concept forms. Users switch their form of concept, but at a lower rate than in previously reported studies.
  - I found that when a spoken dialog system adapts its concept form to the user, the user is significantly more likely to keep his/her originally used concept form.
- A study of language model adaptation to content of user utterances.
  - I built a statistical model to predict which concept is used in the user's utterance. Prosodic and dialog history features are helpful for this prediction.
  - I achieved a statistically significant improvement in speech recognition in a spoken dialog system by adapting the language model to the predicted concept in the user's utterance.

– I achieved a statistically significant improvement in speech recognition in a spoken interface to question answering by adapting the language model to the name in the query.

Before continuing, the reader should be cautioned: this is a computational thesis that uses insights from psycholinguistic studies to improve dialog system performance. My goal is not to build new models of human-human communication, but to be informed by it and to see how these models can be applied to human-compuer communication.

In the next section I outline the contents of the thesis.

1.2 Outline of the Thesis

This thesis consists of eight chapters. My original experiments are described in Chapters 3, 5, 6, and 7.

In **Chapter 2** I review psycholinguistic experiments that look at adaptation in human-human and human-computer conversation, computer science studies of lexical and syntactic adaptation in text and dialog corpora, and spoken dialog systems with adaptation capabilities.

In **Chapter 3** I describe my experiments on measuring adaptation in the human-human Maptask corpus. I introduce two new methods for measuring adaptation and compare adaptation due to recency and due to a specific partner. A preliminary versioin of this work was published in (Stenchikova and Stent, 2007).

Chapter 4 contains a discussion of spoken dialog systems architectures. From a system engineering perspective, I describe how adaptation can be built into modern dialog systems. Part of this work was published in (Stent et al., 2006).

In **Chapter 5** I describe directive adaptation I performed using *Let's Go!* spoken bus information dialog system. I analyze the effect of system prompts on the lexical and syntactic choices of users. I evaluate the impact of prompt design on overall dialog system performance. Part of this work was published in (Stoyanchev and Stent, 2009c).

In **Chapter 6** I describe a responsive adaptation study I performed using a human-computer dialog corpus from the *Let's Go!* dialog system. I evaluate the potential improvement in speech recognition performance of the system adapting its language model to predicted concepts likely to appear in the users utterance. A preliminary version of this work was published in (Stoyanchev and Stent, 2009b).

In **Chapter 7** I describe a responsive adaptation study I performed in the domain of spoken question answering interface. I evaluate the potential speech recognition performance improvements due to the system adapting its language model to the topic of the user's question. I show that responsive adaptation is useful in open-domain QA as well as in closed-domain spoken dialog. A preliminary version of this work was published in (Stoyanchev et al., 2008a).

In **Chapter 8** I summarize my findings and outline future work.

Chapter 2

Previous Research on Adaptation

In this chapter I describe related work in psycholinguistics and in computer science on adaptation in human-human and human-computer dialog.

2.1 Psycholinguistic Perspective

Much prior research on adaptation is done in psycholinguistics. Psycholinguists study language use in dialog by conducting controlled experiments in laboratory settings. *Convergence*, or the evidence of adaptation, is revealed by studies of lexical and syntactic variability in dialog. Researchers find that while there is a great deal of lexical variability across conversations, there is far less lexical variability within a conversation (Brennan and Clark, 1996; Garrod and Doherty, 1994). Brennan (1998) finds evidence for lexical convergence in human-human dialogs and shows that frequency of a word used in a conversation affects the durability of the priming effect for that word.

Psycholinguistic research also aims at identifying the underlying mechanism and causes of convergence in human dialog. Currently, two alternative explanations of adaptation (or convergence) exist in the psycholinguistic literature. One explanation attributes adaptation to *recency* while the other attributes it to *partner adaptation*.

Researchers including Brown and Dell (1987); Pickering and Garrod (2004); and Chartrand and Bargh (1999) attribute adaptation to *recency* (the effect of the most recent utterance). The authors assume a tight coupling of a speaker's mental processes for language production and comprehension. They describe an output/input coordination principle, saying that a speaker formulates an utterance according to words and syntactic rules used to formulate or interpret the most recent utterance(s) spoken and heard by the speaker. Pickering and Garrod (2004) argue that semantic and pragmatic representations used in comprehension are also aligned with those used in production and are evoked through lexical and syntactic priming during interaction. This interactive alignment process also explains why dialog partners may complete each others phrases and reuse the same expressions. Pickering and Garrod (2007) claim that listeners also engage in a production process, predicting the speaker's upcoming words, grammatical categories and meanings: "This emulator enables rapid comprehension and, at the same time, helps listeners deal with noisy input."

In an alternative explanation of adaptation in dialog, researchers including Brennan and Clark (1996); and Horton and Gerrig (2002) argue that convergence and complementarity in dialog is caused by *partner adaptation*. According to the *partner adaptation* theory, speakers build partner models (see Section 1) and adjust their speech production to their current conversational partner. Kraljic and Brennan (2005) show that speakers prosodically mark the boundaries of syntactically ambiguous constructions and adapt their prosody to help their partners disambiguate syntactic ambiguities. Lockridge and Brennan (2002) identify syntactic expressions that dialog participants design specifically for an addressee. Besides explaining *convergence*, *partner adaptation* also explains *complementarity* as adaptation to a dialog partner's needs, where these needs may be different for different dialog participants.

Hartsuiker et al. (2007) evaluate the combined lexical and syntactic convergence. The authors find that syntactic convergence is further enhanced by lexical choices. Speakers are more likely to repeat syntactic forms with the same words than with different words. However, this enhancement effect is short-lived, while the effect of word-independent syntactic priming (syntactic convergence) is long-lasting. The difference in the time frame of pure syntactic and lexicalized syntactic effects points at possible diverse causes of convergence. The authors compare syntactic priming to a form of implicit learning. Reitter and Keller (2007) also hypothesize that short and long-term convergence are evoked by different mechanisms. The authors find that in the short-term dialog participants converge on syntactic constituents but not distituents (part-of-speech pairs that cross constituent boundaries) while in the long-term they converge on distituents as well as constituents.

Regardless of the type or cause of adaptation, researchers have shown that adaptation also occurs when humans interact with spoken dialog systems. For example, users of dialog systems adapt to the system's choice of referring expressions (Brennan, 1996), the system's choice of modality for referring (Bell et al., 2000; Skantze, 2002), and the system's choice of words (Gustafson et al., 1997).

2.2 Measuring Adaptation in Dialog

Computer scientists have also studied adaptation in human-human dialog by analyzing corpora and developing systems with adaptation capabilities. In general, these studies confirm the experimental results summarized above. However, each computational study also produces either (a) an algorithm for measuring adaptation; or (b) an algorithm for modeling/reproducing adaptation in dialog.

Church (2000) introduced a method for measuring lexical 'adaptation'[1] by computing *positive adaptation* using probabilities of co-occurrences. This method determines whether the appearance of a lexical feature in the *priming (earlier) portion* of a document affects the likelihood of its appearance in the *target (later) portion*. The *positive adaptation* for a word w is computed as $Pr(w \epsilon target|w \epsilon prime)$. To determine the *priming effect*, *positive adaptation* is compared to the *prior* $(Pr(w \epsilon target))$. A higher ratio of *positive adaptation* to the *prior probability* indicates a stronger adaptation effect. Church applied this method in a study of a corpus of text documents, treating the first half of each document as the *priming portion* and the second half as the *target*. He showed that positive lexical adaptation does occur, more strongly for content words than for function words. Dubey et al. (2006b) used Church's method to evaluate adaptation for selected syntactic constructions in coordinating structures in the Brown news text and Switchboard dialog corpora. The *priming* and *target portions* in Dubey's coordinating structure experiment were the left and right sides of coordinating constructions (*and* and *or*) in the corpus. The authors reported positive adaptation for each of the syntactic constructions they considered.

In recent work, Reitter et al. (2006b) investigated syntactic adaptation in Switchboard and Maptask. Instead of using Church's method, the authors used logistic regression to examine short-term priming effects within a small window of time in single dialogs. In this method the numbers of occurrences of lexical terms and syntactic constructions are plotted over time after *priming*. This method permits study of the time course of adaptation. A negative slope of a fitted line with a low residual error indicates a priming effect with decay over time. Reitter et al. (2006a) analyzed human-human dialog corpora and detected a strong priming effect for syntactic rules in task oriented dialogs. They also showed rapid degradation of the syntactic priming

[1] Although it was used for measuring adaptation, Church's measure was developed to identify the most useful features for information retrieval, rather than for study of adaptation *per se*.

effect in a dialog over time.

Ward and Litman (2007a) used logistic regression to show that the lexical priming effect in human-human tutoring dialog decays over time. The effect is identified by a negative slope of a line fit to the counts of occurrences of primed words over the dialog. The authors isolated the effect of lexical priming from semantic convergence by removing *no choice* words - words for which there is no alternative synonym (e.g. the, it, is, to). With these words removed, priming effect is still present in the corpus. The authors also measured priming effect on acoustic (energy) and prosodic (pitch) features. While the effect for lexical features is significant, it is very small (slope coefficient<-0.1 for all experiments). The effect for prosodic features is much stronger (slope coefficient -16 for the maximum energy value)[2].

Researchers find that adaptation is related to dialog success. Reitter and Moore (2007) and Nenkova et al. (2008) independently showed that lexical adaptation positively correlates with task success in human-human task-oriented dialog. Ward and Litman (2007b) present evidence that lexical convergence as well as acoustic and prosodic convergence correlate with student learning in tutoring dialogs. The authors show that combined lexical and semantic convergence correlates with student learning even stronger than lexical convergence alone (Ward and Litman, 2008).

In this thesis I devise a new measure for adaptation. The new adaptation measure is a modification of Church's measure. My approach to measuring adaptation differs from previous research as I measure adaptation *between* dialogs. I use the new measure to analyze and compare *partner* and *recency* adaptation. This experiment is described in Chapter 3.

[2] A smaller value indicates a stronger local priming effect.

2.3 Responsive Adaptation in Dialog Systems

I refer to responsive adaptation in automatic dialog as the system's adjustment of its behavior in response to a user or a dialog situation. A system can respond to the user by changing its lexical or syntactic choices (Natural Language Generation), changing its dialog moves (Dialog Management), or adapting its language models or grammars (Speech Recognition/Natural Language Understanding). This section describes several dialog systems that utilize user models to drive adaptation in Natural Language Generation, Dialog Management, or Automatic Speech Recognition. Evaluations conducted by the developers of these systems indicate that responsive adaptation to the user improves system performance.

2.3.1 Natural Language Generation Component

The Adaptive Place Adviser (Thompson et al., 2004) is a personalized spoken dialog system that recommends books, movies, and restaurants. It uses a personalized long-term user model based on domain-specific user preferences for items and item characteristics. The adaptive version of the system learns user preferences from interaction with the user, while the non-adaptive version uses preset preferences. During interaction the system's dialog manager selects a dialog move based on the user's input query, user model, and match of the user's query with a database. Possible system moves include suggesting that the user constrain or relax the query, recommending an item from the database, providing a list of choices, or asking the user for a clarification. In a system evaluation, users interacted with either the adaptive system or a generic recommendation system. System performance, measured by dialog length (in number of turns and time), improved in the adaptive condition.

MATCH (Walker et al., 2004) is a multimodal dialog system for giving suggestions about restaurants. The system tailors every generated utterance to a user model. The user model indicates how important features such as decor, price, and food quality are to the user. The model is static and is generated from an offline user survey. The system selects restaurants to recommend to the user, and determines which attributes of a restaurant to mention, based on the user's query and the user model. In an overhearer-style system evaluation, users were asked to rate the information quality and conciseness of user-tailored and other-tailored system recommendations and comparisons of restaurants. User-tailored recommendations and comparisons were preferred over other-tailored presentations.

Stent et al. (2004) describes a trainable sentence planner for complex information presentations in spoken dialog systems. The syntax of generated system utterances is adapted based on user preferences collected in a rating experiment. An evaluation showed that user-adapted presentations were preferred over presentations generated using templates.

Guo and Stent (2005) show that using individual user preferences for a multi-media presentation results in generation of different presentation styles for different users. User preferences were learned using objective and subjective methods. In the objective method users were asked to reply to questions about the information in the presentation. User preferences were derived from the correctness of their answer which reflected the amount of information retained by the user from the presentation. In the subjective method users were explicitly asked to rate the clarity of the presentation. In the evaluation presentations generated using learned user preferences were ranked as high as presentations generated with manually created presentation styles.

Purver and Kempson (2004) implement a dialog parser/generator using the *priming theory* of Pickering and Garrod (2004) who argue for the interrelatedness of speech production and comprehension processes. In Purver and Kempson's implementation a tree structure represents the semantic interpretation of a string. Parsing and generation use the same tree representations. When a string is parsed, a tree with a corresponding semantic representation is generated. During parsing, trees are built in a word-by-word incremental fashion allowing analysis of anaphora and ellipses. During generation a semantic tree structure is converted into a string. According to *priming theory*, the priming effect causes users to choose the most recently used syntactic rule or word during production. In Purver and Kempson's implementation the lexicon search considers the most recently used words first, emulating lexical adaptation. The incremental nature of the generation process also allows the generation of bare fragments reusing structure from previous sentences, e.g. *What did you eat for breakfast? Porridge.*

2.3.2 Dialog Manager Component

A dialog manager is equivalent to the brain of a dialog system. At each point in a dialog, the dialog manager is responsible for choosing the system's next action. Adaptation in the dialog manager affects the system's choice of actions and dialog acts. Rules for adaptation in a dialog manager can be manually encoded or automatically learned.

Brennan and Hulteen (1995) apply a collaborative theory of human communication theory (Clark and Schaefer, 1989) to human-computer communication. They aim at efficiency in dialog by providing just enough evidence to the user to handle a system's error. Their experimental system dynamically adjusts its grounding criterion and adapts the amount of feedback given to the user of the dialog system. Rules based on the dialog history, the physical environment, and the task model are used to determine what kind of feedback messages to provide and when to provide them. The proposed model filters out the excessive feedback that in a Wizard-of-Oz study users found annoying, but leaves the feedback that users find important.

Another example of a dialog system where adaptation is encoded in a set of rules is described by Komatani (2005). The system generates cooperative help messages based on a user model. The user model includes the skill level, knowledge level, and urgency of the user. The user model is determined automatically at the time of the user's call based on dialog-initial user utterances. Evaluation results show that novice users learn to communicate with the system more efficiently when the user model is utilized while more advanced users are not forced to listen to basic help messages and experience shorter completion times.

Recent work on dialog modeling explores reinforcement learning for automatic determination of dialog moves (Bohus et al., 2006; Williams et al., 2007; Lemon et al., 2006; Henderson et al., 2006). These approaches are adaptive to the dialog situation: the choice of system action is based on the user's previous actions. Bohus et al. (2006) automatically learn to choose an error recovery policy between explicit confirmation, implicit confirmation, and an extensive help message. Williams et al. (2007) improves a handcrafted dialog manager by incorporating reinforcement learning. The authors explicitly encode a user goal model that tracks how the user's goal changes over time.

Komatani et al. (2007) describe an approach to presenting users with help after misrecognition that is adapted to the current status of information obtained by the system. Their dialog system tracks the *known degrees* of each node in a system domain concept tree. The domain concept tree contains four layers: system, function, element, and content word. The *known degrees* are updated after each user utterance. For example, if the user utters a content word, the *known degrees* of the corresponding concept in the content layer is incremented. This method allows the system to selectively present help messages only when they are necessary.

2.3.3 Automatic Speech Recognition Component

Speech recognition is one of the largest causes of errors in human-computer dialog. Although domain-dependent speech recognition in a dialog system is more tractable than open-domain speech recognition, user variation in grammar and vocabulary causes problems for robust speech recognition. In the past twenty years, since the breakthrough in basic speech recognition (Rabiner and Juang, 1986), researchers have been working on incremental improvements to the Hidden Markov Model (HMM)-based algorithm. In recent years researchers have focused on: improving statistical language modeling technology; combining grammar-based ASR with statistical language modeling; and giving users additional guidance about the language the system can process. Acoustic and prosodic variations due to hyperarticulation in dialog have been investigated in Soltau (2005).

Language Model Adaptation

A language model encodes probabilities of n-grams (strings of length n) occurring in an utterance. The similarity between the user utterance and the dataset used for generating a language model affects the performance of recognition. For example, an utterance containing words frequent in the language model is more likely to be recognized correctly than an utterance containing infrequent words.[3] Language model adaptation is a technique for improving speech recognition. It involves adjusting probabilities in the language model or selecting the data for building the model. The goal of this adaptation is to make the model more similar to the data, leading to speech recognition improvement.

Riccardi and Bangalore (1998) describe an improvement to a system's language model by learning phrase grammars with unsupervised clustering techniques (iterative entropy reduction). They automatically learned phrase grammars allow for generalization. Using these grammars one may automatically generate phrases never seen in the training corpus, yet similar to the phrases in the training corpus. Evaluation shows improvement in a call classification task.

[3] Please refer to Chapter 6 for further description of language model use in 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 dialog system, leading to reductions in word error rate. It has now become standard practice to use dialog state specific language models (Bechet et al., 2004). Depending on the dialog state, the language model gives more weight to the words and phrases that are more likely to be used in that state.

Iyer and Ostendorf (1999) adapted the language model based on topic rather than on dialog state. They obtained a 4.5% reduction in word error rate on the Wall Street Journal text corpus by using a weighted combination of topic-specific language models, but only a 1.2% relative reduction in word error rate on the Switchboard spoken dialog corpus. An example of language model use in natural language understanding is Dubey et al. (2006a), who incorporate probability models of previous syntactic rule use into an incremental parser. Incremental adaptive parsing can potentially be useful for a dialog system if user utterances are recognized incrementally.

Co-constraining Automatic Speech Recognition and Natural Language Understanding (NLU) has been shown to benefit both processes. Young (1994) use output from the NLU along with acoustic probabilities to detect misrecognized words on a second pass through the recognizer.

Using Grammar-Based Speech Recognition

Grammar-based and statistical ASR have been combined in numerous previous research projects to improve ASR 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.

Gabsdil and Lemon (2004) implement context-sensitive speech recognition by using machine learning on a combination of acoustic and dialog context features. They automatically learn rules for grammar switching in a dialog system.

In my experiments I also address the speech recognition problem through language model adaptation using a 2-stage recognition approach. The language model adaptation methods described above introduce context sensitivity into dialog systems. In my experiments, I increase context sensitivity in an already context sensitive dialog system. I use machine learning on features similar to these used by Gabsdil and Lemon (2004) and Litman et al. (2006). The novelty of my approach is the type of information that I predict using machine learning: task-related concepts in the user utterance. I adapt the language model to the expected concepts in the user utterance and achieve improvement in speech recognition performance. My speech recognition experiment with a dialog system is described in Chapter 6. I extend this work to open-domain speech recognition experiment with a spoken interface to question answering, as described in Chapter 7.

2.4 Directive Adaptation

I refer to directive adaptation as system utterances or actions that guide users to change their behavior. In this section I include descriptions of studies of the effect on user behavior of various types of directed help messages, the user's initial perceptions about the system, and the system interaction style, such as personal/impersonal or polite/impolite.

2.4.1 Directive Help Messages

Systems often provide help information to users when users have trouble communicating with the system. Recently researchers have experimented with providing additional guidance to users with the specific goal of shaping their responses. For example, Tomko and Rosenfeld (2006) "teach" users to utilize "speech graffiti", a limited language of simple commands, when speaking with the system. The authors find that in the movie theater information domain, their *limited input* system outperforms a *flexible input* one.

In the movie theater domain the number of types of different user questions is relatively small. However, in a call routing domain, where the number of possible user utterances is large, Sheeder and Balogh (2003) find a higher call routing accuracy when users speak with natural language sentences rather than keywords. Furthermore, the authors find an effect of system help message type (keyword vs. natural) on the immediately following user utterance command. They find that a *natural* help message causes users to also use *natural* sentences in their input and improves system performance.

Gorrell et al. (2002); Hockey et al. (2003); and Fukubayashi et al. (2006) look at targeted help system messages following misrecognition errors. Gorrell describes two approaches where automatically recognized user utterances with low recognition confidence scores are used to identify a help message. In the first approach the help message is one of several predefined messages. A classifier using features from the speech recognizer selects an appropriate help message to be played to the user. In the second approach a help message is generated by matching the user's utterance to the closest in-grammar utterance. The first approach was implemented in a command and control application and the second in a question answering application. Both applications were evaluated and lead to significant improvements in task completion rates and user satisfaction.

Hockey et al. (2003) assists users in becoming experts by providing informative help messages. The researchers identified three major types of errors in their push-to-talk command and control application: 1) endpointing errors (when a initial word is cut off), 2) out-of-vocabulary words, and 3) subcategorization errors (in-vocabulary but out-of-grammar). The authors designed rules to handle each type of error. For example, after an out-of-vocabulary error the system response is "the system does not understand the word X".

Rotaru and Litman (2006) examine a human-computer tutoring dialog corpus and identify that emotions and certainty interact with speech recognition problems in the system. Forbes-Riley and Litman (2009) show that incorporating the information about the user's certainty into system response strategies improves tutoring dialogs.

2.4.2 System Interaction Style

Brennan (1991) find that both users' system models and systems' responses have an effect on users' syntactic and dialog act choices. For example, in a Wizard-of-Oz study, she finds that users are more likely to acknowledge a response when they believe that the partner is a human rather than a computer. Independent of the user's system model, the user adapts to the style (long vs. short) of the preceding system utterance. Brennan and Ohaeri (1994) compare an anthropomorphic text dialog system (that refers to itself using a personal pronoun "I") and a non-anthropomorphic system. The authors find that users are more likely to use personal pronouns with an anthropomorphic system. Kruijff-Korbayova and Kukina (2008) confirm this finding in a spoken dialog system. Both studies find no significant effect of the user's perception of the system's level of proficiency on the user's use of personal pronouns . By contrast, Pearson et al. (2006) use an experimental setup to show that the strength of user adaptation effect is determined by the user's perception of the system's level of proficiency, which can be manipulated by a single 10 second screen display prior to the start of the dialog.

In my work I evaluate whether syntactic and lexical choices in system prompts affect the user choices in his/her responses. I confirm some of the results described above in a deployed system with real users. I also explicitly address adaptation to the form of task-related concepts. These experiments are described in Chapter 5.

Chapter 3

Measuring Adaptation Between Dialogs

3.1 Motivation and Research Goals

In this chapter I describe an adaptation study of human-human dialog. The work described here builds on the psycholinguistic and computational studies described in Sections 2.1 and 2.2. My research aims at modeling humans' lexical and syntactic choices during speech production.

As I said in previous chapters, currently there is a debate in the psycholinguistics community about the causes of adaptation, whether adaptation is:

- *partner adaptation* – adaptation based on a model of the partner (Brennan and Clark, 1996; Horton and Gerrig, 2002).

- *recency adaptation* – adaptation due to representations of words, concepts etc. being *activated*, or brought to the forefront during language production, by previous perception or comprehension (Brown and Dell, 1987; Pickering and Garrod, 2004; Chartrand and Bargh, 1999).

In this work I address the questions outlined in the Table 3.1.

1	Can we identify the features that affect partner adaptation and recency adaptation?
2	Is partner adaptation or recency adaptation more prevalent?
3	Does feature frequency in the *prime* affect feature frequency in the *target* ?

Table 3.1: Questions addressed in adaptation study

Existing measures of adaptation previously introduced by other researchers (Church, 2000; Reitter et al., 2006b) do not directly permit separation of adaptation due to the partner or to recency. Also, neither of the existing measures examines how frequency of a feature w in the *prime* affects the likelihood of consecutive occurrence of this feature (I call this *adaptation strength*[1]).

To address these issues I propose two new measures, one that measures the presence of adaptation and another that measures its strength. Together, these measures can identify adaptation within a single document or between documents; can identify the strength of adaptation as well as its presence; and can be used to identify the source of the adaptation. I use these measures to study adaptation in the Maptask spoken dialog corpus. I close this chapter with some ideas about how to apply these measures to dialog system development, and some ideas for future work.

3.2 Experimental Method

3.2.1 Previous Adaptation Measures

Two adaptation measures introduced by Church (2000) and Reitter et al. (2006b) are found in the literature. These measures analyze how use of a feature (or a linguistic stimulus) affects its consecutive occurrences. Both of the measures assume that the

[1] This is different from *strength* discussed by Reitter

prime portion of a document exhibits a linguistic stimulus. The measures analyze the effect of this stimulus in the *target* portion of the document. Both of the measures have been applied for evaluating lexical and syntactic adaptation in text and dialog. Although Church designed adaptation measure to analyze feature occurrences in text documents with a goal of improving information retrieval, his measure is also suited for analyzing adaptation in dialog. Dubey et al. (2006b) consequently used Church's measure to analyze adaptation in dialogs. In this work, I start with Church's measure and change it for the purpose of measuring adaptation between dialogs.

In Church's measure the *prime* and *target* are two separate partitions of a document (see Figure 3.1).

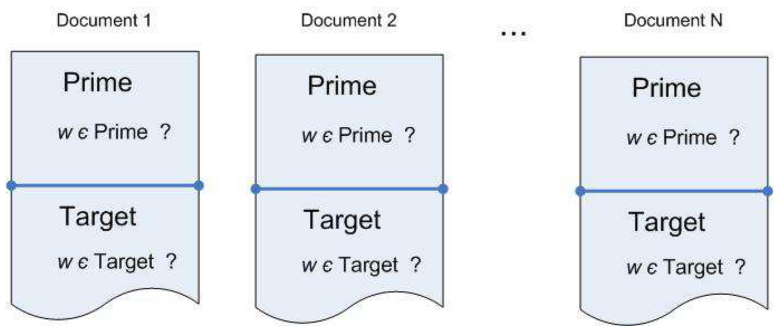

Figure 3.1: Church's adaptation model

Church computes adaptation for each feature over a set of documents. A stimulus for a feature w exists in a document if w is present in the *prime* partition. The probability of positive adaptation is computed as a function of occurrences of a feature w in the *prime* and *target* partitions:

$$P_{+}(w) = Pr(w \in target \mid w \in Prime) = \frac{w_{p,t}}{w_{p,t} + w_{p,\bar{t}}} \tag{3.1}$$

(See notation explanations in Table 3.2.) Positive adaptation is compared to a

notation	explanation
$w_{p,t}$	# prime/ptarget pairs where w occurs BOTH in prime and in target
$w_{p,\bar{t}}$	# prime/ptarget pairs where w occurs in prime and NOT in target
$w_{\bar{p},t}$	# prime/ptarget pairs where w occurs NOT in prime and in target
$w_{\bar{p},\bar{t}}$	# prime/ptarget pairs where w occurs NEITHER in prime NOR in target

Table 3.2: Notations for Church's adaptation measure

prior, the probability of a feature w occurring in the target:

$$P_{prior}(w) = Pr(w \in target) = \frac{w_{p,t} + w_{\bar{p},t}}{N} \tag{3.2}$$

A higher ratio between positive adaptation for a feature w and the prior for this feature indicates stronger adaptation for the feature. Church's measure allows to compare the adaptation effect between different features.

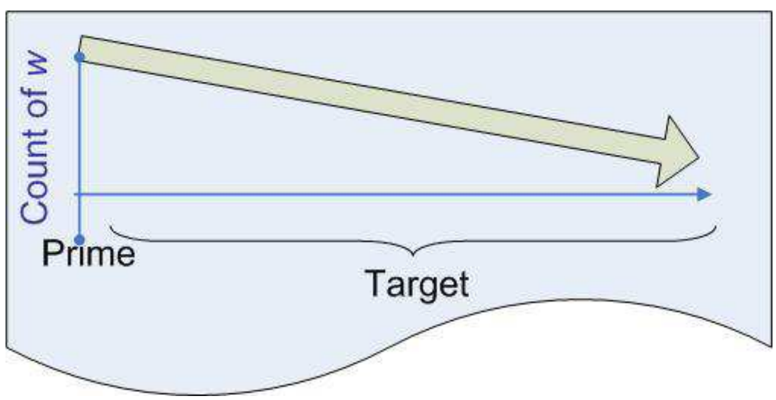

Figure 3.2: Reitter's adaptation model

Another measure, introduced by Reitter et al. (2006b), evaluates *priming* over time in a document or dialog. This approach uses a sliding window where each sentence (or utterance) is considered a *prime* containing a stimulus and consecutive sentences are considered the *target*. The sliding window is applied throughout the document. Linear regression approximates a linear relation for the number of the repetitions of a feature over the distance from its *prime* (see Figure 3.2). The slope of the fitted line reflects the effect of the *prime*. A negative slope of the approximated line indicates priming effect and its decay over time. A steeper slope indicates a stronger *priming* effect. Reitter's method allows to study the time course of adaptation and detect attenuation over time. Ward and Litman (2007a,b, 2008) applied this measure to examine priming in tutoring dialogs.

In my study of adaptation I compare *recency* and *partner* adaptation and address the research questions outlined in Table 3.1. I evaluate adaptation *between dialogs*. Adaptation due to recency studied in this work differs from the priming effect studied in (Reitter et al., 2006b), because I hypothesize that there is adaptation between dialogs affecting a speaker's linguistic choices.

Let's assume that three speakers Ali, Bob, and Tom take turns participating in dialogs (see Figure 3.3). First, Ali talks to Bob, then Ali talks to Tom, then Ali talks

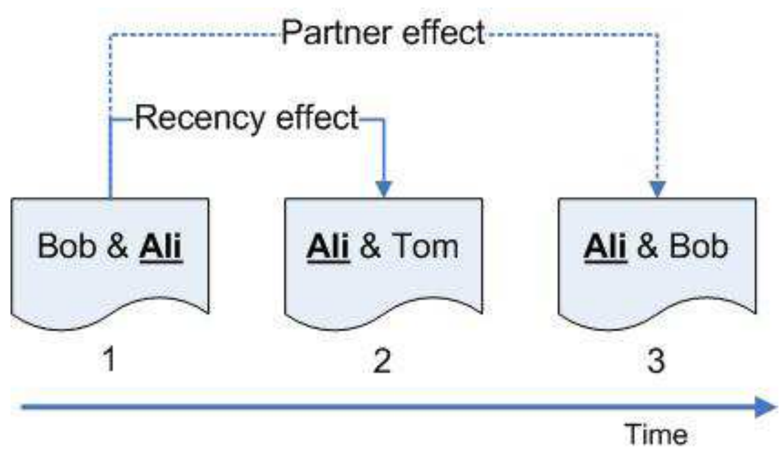

Figure 3.3: Comparing Partner and Recency adaptation effect

to Bob again. The first dialog is the *priming* dialog (Ali is primed by Bob). The second dialog is the *target* where I look for adaptation due to recency. The third conversation is the *target* where I look for adaptation due to the partner.[2] I measure

[2]The second dialog with Tom plays the role of distructor for measuring partner effect in the third

the effect of priming by Bob on Ali's utterances and examine the effect in the second and the third dialogs. A bigger adaptation effect in the second dialog is an argument for adaptation due to recency, while a bigger adaptation effect in the third dialog is an argument for adaptation due to the partner. The effects are compared by the number of features exhibiting adaptation and by the *strength* of this adaptation (the proportionate increase in frequency of a feature in the *target*). Partners in a single dialog may also have an effect on each other's utterances. However, in this study I measure adaptation between dialogs. For example, I am not measuring the effect of Tom on Ali in the second dialog although this effect is likely to be present.

My choice of the adaptation measures for this experiment is guided by the goals of the experiment. In my experiment the *prime* and *target* pair are two separate dialogs. I hypothesize that lexical and syntactic choices in the *prime* dialog have an effect in the *target*. Hence, Reitter's measure of attenuation in priming over time is not applicable to test my hypothesis. Church's measure was developed to identify the most useful features for information retrieval, rather than for study of adaptation *per se*. Consequently, it has several disadvantages for studying adaptation directly:

- For each feature, his method provides an answer to the question "Did the feature occur in the prime/target?"; however, it does not take into account the frequency of occurrence of a feature, so it cannot be used to measure the effect of frequency in *prime*.

- His method requires large amounts of data in order to obtain a statistically significant support for the hypotheses. It cannot be used to identify adaptation in a single document or between a single pair of documents.

- His method under-reports adaptation in frequently occurring features, such as closed class words, that are present in essentially every document.

---

dialog.

3.2.2 Proposed Adaptation Measures

I propose two measures. The first one measures the prevalence of adaptation between two documents, while the second one measures the strength of adaptation. The measures are outlined in Table 3.3.

Measure	Description
Adaptation Ratio	Measures presence of adaptation: priming of a feature increases its probability in the target
Adaptation Strength	A feature exhibits *stronger* adaptation if it is more frequent in the target

Table 3.3: Adaptation measures

Throughout this discussion, I will use the term *document* to refer to a dialog or part of a dialog, and the term *feature* to refer to any phenomenon (lexical, syntactic, referring expression, dialog act, etc.) that occurs in or is labeled in a dialog.

To measure the degree to which a feature f exhibits adaptation, following Church's approach I divide the corpus into a collection of *prime* documents and *target* documents. In Church's document the *prime* and *target* were separate portions of the same document while in my approach the *prime* and *target* are two different documents. For each feature f, I compute the frequency of occurrence of the feature in the *prime* document (p), the *target* document (t), and the corpus as a whole (baseline, or b). I chose to use all documents in order to have a larger corpus for the estimation. Assuming that priming effect is present, *target* documents may have higher frequencies of the primed features. By including *target* documents my baseline values may be higher leading to more conservative measures. One may use relative frequencies rather than absolute frequencies, or smooth low-frequency features; I do not do this in the experiments reported in this study because earlier experiments showed that these did not change my results. Both of my measures compare p and t to b. I use

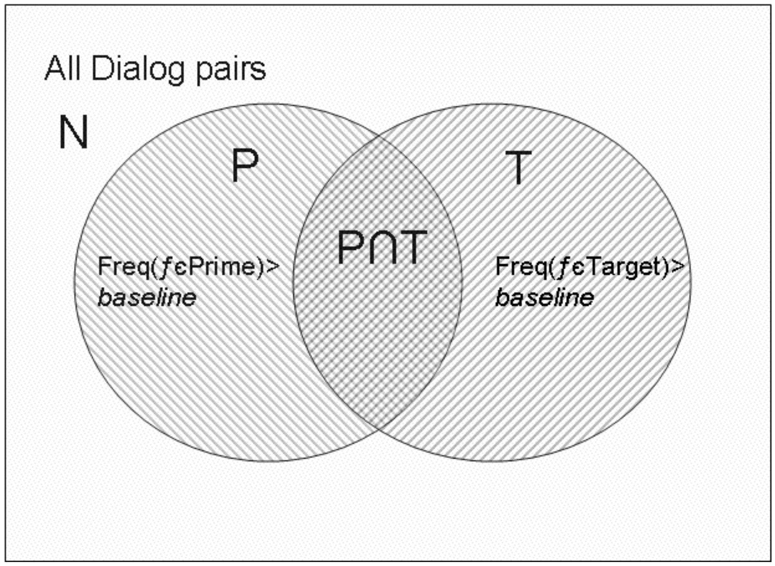

Figure 3.4: *Prime* and *target* dialog pairs used in my adaptation measure. N is the total number of (prime, target) dialog pairs; P is the number of prime dialogs where frequency of $f > b$; T is the number of target dialogs where frequency of $f > b$; P∩T is the number of prime&target dialog pairs where frequency of $f > b$ in both prime and target dialog.

the notation $f \in D$ as a shortcut to indicate that the frequency of occurrence of f in document D is greater than the baseline frequency for f.

3.2.3 Adaptation Ratio Measure

The adaptation ratio measures presence of adaptation for a feature in a set of *prime* and *target* document pairs. Similarly to Church's measure, *adaptation ratio* for a feature is computed as a ratio of the probability that the feature occurs after priming (+adapt) to the probability that it occurs by chance.

$$AdaptationRatio = +adapt/chance \tag{3.3}$$

However, my methods of computing *+adapt* and *chance* differ from Church's.

This measure is a modification of Church's measure in two ways. First, it uses the frequency of occurrence of a feature in each document rather than merely its presence or absence. For each feature, I compute a baseline b – the average frequency of the feature per document. I consider a feature to be *primed* in a *prime/target* document pair if its frequency in the *prime* document is greater than b. I consider a feature to be *adapted* if its frequency in the *target* document is greater than b. To compute adaptation, I use counts of prime and target documents where frequency of a feature is above the baseline (P, T, P∩T in the Figure 3.4). Using Church's measure, a frequent feature, such as "the", is *primed* in almost all document pairs, while the *adaptation ratio* measure considers it *primed* only if its frequency is above the baseline. The *adaptation ratio* measure evaluates adaptation both on frequent features and on infrequent features. Second, instead of using Church's prior probability I use an estimate of the probability of feature co-occurrence in prime and target by *chance*. Next, I describe how *chance* and *+adapt* probabilities are calculated. See Figure 3.4 for a graphical illustration of the parameters used in my calculations.

Chance

The probability of a feature co-occurring in prime and target by chance is the product of probabilities of its occurrence in prime and target independently, assuming independence of the two. It is represented by $P \cap T$ area in the Figure 3.4.

$$P \cap T = Pr(f \in prime \cap f \in target) =$$

$$Pr(f \in prime) * Pr(f \in target) \tag{3.4}$$

For N $(prime, target)$ dialog pairs where feature f occurs more that b times in P $primes$ and more than b times in T $targets$, the probability of chance co-occurrence of f in $prime$ and $target$ can be approximated by:

$$chance = (P/N) * (T/N) \tag{3.5}$$

+Adapt

Church defines positive adaptation for a feature f as follows:

$$+adapt = Pr(f \in target \mid f \in Prime) \tag{3.6}$$

which I approximate as:

$$+adapt = P \cap T/P \tag{3.7}$$

I compute for each feature both *chance* and *+adapt*. I define the *adaptation ratio* as *+adapt/chance*. I sort the features in decreasing order by adaptation ratio. Those at the top of the list exhibit more positive adaptation. I also compute χ^2 to identify features for which the adaptation ratio is significant.

3.2.4 Adaptation Strength Measure

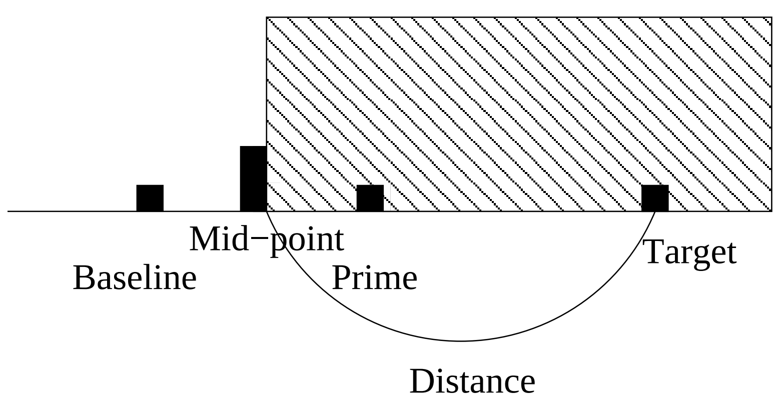

Figure 3.5: Distance adaptation measure

The adaptation Strength measure identifies the effect of priming on the feature frequency in the target. I use this measure to analyze the relation between the frequencies of a feature in the prime and target. Instead of using binary values for each feature indicating presence or absence of that feature in a document, I use the actual frequency of occurrence of the feature in the document. I assume that a feature exhibits *stronger* adaptation if it is more frequent in the target.

To measure the strength of adaptation on a per-feature basis, I use a *distance* measure. For a feature f with frequency in *prime* of p, frequency in *target* of t and baseline frequency b,

$$distance = t - (p + b)/2 \tag{3.8}$$

Imagine adaptation as a force pulling t towards p and away from b. If there is positive adaptation, then t will be closer to p than to b, as illustrated in Figure 3.5. I consider a feature to be *adapted* in a pair of dialogs if the target point lies to the right of mid-point in the figure (I conservatively chose the midpoint between b and p; a point closer to b could be chosen for a more liberal interpretation of adaptation). *Distance* is computed for each feature for each dialog pair. Its value suggests the strength of adaptation for this feature in this dialog pair. I define by the *adaptation strength* for a dialog the average *distance* over all *adapted* features.

3.3 Data

The Maptask corpus (Anderson et al., 1991) contains 32 sequences of dialogs involving four speakers who discuss routes displayed on maps and trade dialog partners as shown in Table 3.4. In each dialog, one partner is a *giver* of the route description and the other is a *follower*. From each of the 32 Maptask dialog sequences, I extract the dialog triples (1,4,6) and (2,3,5) corresponding to the sequences of dialogs between Ali, Bob, and Tom in Figure 3.3. The *follower* in the first dialog of each triple, Ali, is

dlg #	giver	follower	pair1	pair2
1	a1	b1	*prime* (Bob&Ali)	
2	b2	a2		*prime* (Bob&Ali)
3	a2	a1		*recency* (Ali&Tom)
4	b1	b2	*recency* (Ali&Tom)	
5	a2	b2		*partner*(Ali&Bob)
6	b1	a1	*partner* (Ali&Bob)	
7	a1	a2		
8	b2	b1		

Table 3.4: Maptask dialog order

the *giver* in the second and third dialogs. In the second dialog, Ali speaks with a new partner Tom. In the third dialog Ali speaks with Bob, the same partner as in the first dialog. I hypothesize that recency adaptation by Ali may be displayed in the second dialog in each triple (Ali&Tom), which are the next dialogs after priming for Ali, and partner adaptation of Ali to Bob may be displayed in the third dialog in each triple (Ali&Bob), which are the dialogs with a repeated partner for Ali. To evaluate the recency effect, I use *giver* Bob's utterances in the first dialog as the *prime* document and *giver* Ali's utterances in the second dialog as the *target*. To evaluate the partner effect, I use *giver* Bob's utterances in the first dialog as the *prime* and *giver* Ali's utterances in the third dialog as the *target*. This analysis allows me to compare how the effect of the first conversation with Bob differs between Ali's later conversations with Tom and with Bob. In the following sections I will refer to the first and second dialogs as the *recency* dialog pair and to the first and third dialogs as the *partner* dialog pair.[3]

[3]It would also be interesting to measure adaptation if the dialog with the same partner (Ali & Bob) immediately followed the priming dialog (Bob & Ali), but Maptask did not have this scenario.

3.4 Experiments

I consider two feature types: lexical (word stems, part-of-speech tagged to help distinguish between word senses; and bigrams); and syntactic (productions from the Maptask parse tree annotations).

3.4.1 Identifying Adapting Features

In this experiment I use the *adaptation ratio* and χ^2 test to identify features that are affected by priming. I conservatively define an *adapted feature* as a feature with:

1. *Adaptation ratio* +adapt/chance > 1

2. Occurring in more than 10 prime dialogs (to enable statistical inference) with frequency higher than the baseline

3. χ^2 significance level above 95%

Adaptation ratio> 1 identifies features that are likely to become more frequent after being primed.

χ^2 measures the statistical significance of the priming effect. χ^2 is an approximation and is applicable only when the values of the variables are "large". Some statistics books consider this number to be >5 and others >10. Hence, I chose a threshold of 10 for the frequency of a feature in *priming* documents. I use both the *adaptation ratio* and χ^2 measures in conjunction to increase the accuracy of finding adapting features.

Table 3.5 shows examples of two stem/POS features and their *+adapt* and *chance* values. The feature *you/DET* occurs with frequency above the baseline in 13 (out of 32) priming dialogs. It occurs 8 times with frequency above the baseline in both prime and target dialogs. For this feature, *+adapt* is .62 and *chance* is .14. The

+adapt/chance ratio is 4.4 > 1 and χ^2 is above 4.841, the 95% significance level for a test with one degree of freedom. Hence *you/DET* exhibits adaptation. For the feature *finish/VB +adapt* is less than *chance*. Hence, *finish/VB* does not exhibit adaptation.

feature	prime	target	prime $\cap$target	+adapt	chance	$\frac{+\text{adapt}}{\text{chance}}$	χ^2
you/DET	13	11	8	0.62	0.14	4.4	7.16
finish/VB	11	9	1	0.09	0.10	0.9	3.00

Table 3.5: Example lexical features with adaptation ratio and chi^2. Significance level is >95% when $\chi^2 > 4.841$

	partner	recency
ADJ	right-hand	bottom, right-hand
ADV	when, diagonal	right, well, about
AUX		have
CONJ	if	till, that, so
DET	you, across, on, what, that	my, i, just, that
INTJ	sorri, er,	uh
NOUN	bottom	map
PREP	across, through, along, from	from, by, to
VERB	know, got, take, pass	say

Table 3.6: *Adapted* word-stem features

Tables 3.6 and 3.7 show the *adapted* stem/POS and bigram features. I observe two interesting lexical categories of features that adapt: perspective and directionality. In Maptask, speakers can take up a "map-based" perspective (and use words like *north, south, east, west*) or a "paper-based" perspective (and use words like *right, left, top, bottom*). For example, if Bob in the first dialog of Figure 3.3 said *right-hand* more frequently than average, Ali in both the second and third dialogs is likely to say *right-hand* more frequently than average. Lexical features indicating perspective are adapted in both partner and recency dialog pairs; the same is true for bigram

type	partner	recency
Perspective & Directionality	about the, abov the, just abov, right-hand side, round the, up toward, your left	down about, down to, just below
Motion	come to, you come, go round	
With pronoun I/my		i mean, my map, on my, yeah i
With pronoun you	if you, right you, when you, you got, you just, your left	now you, no you, you just, you got, 'til you
With *no*		no no, no you
Other	a wee	just to, okay and

Table 3.7: *Adapted* bigram features

features. Other features in this category (e.g. *left, top*) also show adaptation but occur too infrequently for the adaptation to be significant. Directionality in Maptask is indicated by prepositions such as *across, through, along, around, from, by, to.* These prepositions are intimately tied to the spatial perspective by a pair of conversation partners. Most of them are adapted for partner dialog pairs.

Verbs and bigrams containing motion verbs such as *come, go, got, take,* and *pass* are adapted in partner dialog pairs but not in recency dialog pairs. This finding suggests that Ali adapts to her partner Bob in the usage of motion verbs. If Bob in the first conversation uses motion verbs more frequently than average, Ali does not tend to increase her usage of these verbs in the next conversation with a different speaker Tom. However, when speaking again with Bob she is more likely to use these words more frequently.

Table 3.8 shows the adaptation ratio and adaptation strength for the selected features with the highest difference between partner and recency adaptation ratios. All of these features (except *to be*) have adaptation ratio >1 (although not all are significant according to the χ^2 test). Directional features *across* and *through* have a higher partner adaptation ratio than recency adaptation ratio. However, adaptation

feature	Adapt. ratio		Adapt. strength	
	partner	recency	partner	recency
across	**7.314**	4.655	0.285	**3.452**
through	**5.642**	3.385	0.785	**1.285**
sorri	**4.180**	1.741	**0.410**	0.161
i	1.714	**3.0**	7.240	**8.573**
uh	3.413	**5.973**	**1.054**	0.471
sai	1.693	**5.642**	**2.430**	4.680
about the	**4.478**	1.492	0.640	**2.016**
right-hand side	**5.924**	3.022	**2.099**	1.640
when you	**5.642**	2.987	**0.660**	0.493
my map	2.418	**7.052**	**1.816**	0.416
on my	3.173	**6.770**	**1.328**	0.328
to be	0.846	**3.847**	0.265	**1.065**

Table 3.8: Comparing *adaptation ratio* and *adaptation strength* in partner and recency dialog pairs for the features with the highest differences between the adaptation ratios. The highest score in each pair between partner and recency is highlighted in **bold**.

strength is higher for recency dialog pairs. This means that if Bob in the first conversation said *across* more frequently than average, Ali will be likely to say *across* more frequently in both of the the following conversations with Tom and Bob, but more so in the conversation with Bob. However, if she did say *across* while talking to Tom, she would be more likely to say it more frequently than in the conversation with Bob.

Bigram and word-stem features containing a first person pronoun (*I/my*) exhibit statistically significant adaptation for recency pairs but not for partner pairs (Tables 3.6 and 3.7). If Bob in the first conversation uses more than average first person pronouns, Ali is affected in the next conversation with Tom by also using more than average first person pronouns. However, Ali will not be affected by the priming for first person pronouns when she speaks again with Bob in the third dialog. The features *I*, *my map*, *on my* are among the features with the highest differences between partner and recency adaptation ratios in favor of recency (Table 3.8). Bigrams and word-stem features containing a second person personal pronoun (*you*) exhibit both partner and recency adaptation. In the Maptask domain, one speaker describes directions using a map while the other speaker follows the directions on a different map. The direction giver can use different strategies of achieving the task. One of the strategies is to describe what speaker sees on his or her own map, in which case the speaker would use the first person pronoun. Another strategy is to ask or try to guess what is shown on their partner's map, in which case the speaker would use second person pronoun. The adaptation in individualistic pronouns suggests that speakers adapt game strategies to their partners. The game strategy where the speaker describes his or her own map is adapted to the most recent conversation partner, while the game strategy where the speaker asks about the other person's map is adapted both to the most recent conversation and to the specific partner.

Bigrams containing a negation (*no*) are adapted in recency dialog pairs but not in partner dialog pairs. If Bob was excessively negative in the first dialog, Ali will be negative in the next dialog with Tom but not when she speaks again with Bob. The effect of priming of negation is similar to the effect of priming of the individualistic pronouns. Negation can be described as another strategy where the speaker wants to go back to

	partner	recency
advp→		advp
np→	at at ap nn	ap nn; np ap nn; at nn nn; np; np np; pn; ppg nn
pp→	in; rp	pp not pp; ql rp pp; rp aff
s→	s aff aff s; hv np vp; np; np bez; s s	aff s; np; np s
vp→	vp be np; bez pp; to vp; vb np pp; vb vb pp; vbg pp	advp vp; ber vp; md vp; vb np; vbg; vbg pp vbn pp; vp vp

Table 3.9: Examples of *adapted* syntactic features

Overall more syntactic features exhibit statistically significant recency adaptation than partner adaptation (see Table 3.9). For the noun phrase syntactic productions (NP) only one production exhibits statistically significant partner adaptation

(NP→AT[4] AT AP[5] NN), while seven productions exhibit recency adaptation. For the prepositional phrase syntactic structure (PP) I find both partner and recency adaptation but the rules exhibiting adaptation differ between partner and recency dialog pairs. The structures where PP is a single preposition: PP→IN[6] and PP→RP[7] exhibit partner adaptation, while the structures PP→PP NOT PP, PP→QL RP PP, and PP→RP AFF exhibit recency adaptation. For the syntactic sentence structures (S) complex sentence structures, such as S→S AFF AFF S and S→S S exhibit partner adaptation, while simple sentence structure S→NP exhibit both partner and recency adaptation. Sentence structures starting with the auxiliary verb *have* S→HV NP VP or ending with the auxiliary verb *is/was* S→NP BEZ[8] exhibit partner but not recency adaptation. For the verb phrase structures (VP) I find both partner and recency adaptation but the structures exhibiting partner and recency adaptation differ. For example, the structure with the auxiliary verb *is/was* VP→BEZ PP exhibits partner but not recency adaptation.

feature	Adapt. ratio		Adapt. strength	
	partner	recency	partner	recency
NP→NP PP	1.896	2.6	31.699	17.249
NP→NN	2.963	2.963	0.781	2.656
NP→DT NN	3.048	3.048	0.445	0.695
NP→DT AP NN	2.308	3.077	0.254	0.503

Table 3.10: Adaptation for the syntactic features examined by Dubey

Table 3.10 shows adaptation ratio and adaptation strength for some of the syntactic features that were examined in Dubey et al. (2006b). Dubey found small but statistically significant adaptation for these features. In my experiment, the features

[4]AT includes articles *a, an, no, the*.

[5]AP includes *few, further, final, last, least, less, little, many, more, most, much, next, only, other, same, single, very*

[6]IN includes prepositions

[7]RP include adverbial prepositions

[8]BEZ includes auxiliary verbs *is, was*

have adaptation ratio above 1 but the effect is not statistically significant according to the χ^2 test. All but the first and last features show comparable partner and recency adaptation ratios. The adaptation strength for the feature NP→NP PP shows stronger partner adaptation than recency adaptation. By contrast, the feature NP→NN shows stronger recency adaptation.

3.4.2 Comparing Partner and Recency Adaptation

In this experiment, I use *adaptation ratio* and *adaptation strength* to compare partner and recency adaptation. Table 3.11 shows *adaptation ratio* and *adaptation strength* averaged over all features for each feature type (Stem/POS, Stem/bigram, Syntactic).

According to the *adaptation ratio* measure, there is no significant differences between partner adaptation and recency adaptation for lexical features. However, according to the *adaptation strength* measure, lexical features (Stem/bigram) have stronger adaptation in the recency dialog pairs. Syntactic features, taken as a whole, have significantly higher *adaptation ratios* for recency than for partner.

Table 3.12 reports the same measures as Table 3.11 over the subset of *adapted* features from Tables 3.6, 3.7, and 3.9. The results on this subset of *adapted* features are similar to the results for all features.

feature	Adaptation ratio		Adaptation strength	
	partner	recency	partner	recency
Stem/POS	2.64	2.71	3.46	3.67*
Stem/bigram	2.99	3.03	1.71	1.91*
Syntactic	2.71	2.92*	4.70*	4.11

Table 3.11: Average adaptation ratio and adaptation strength for *all* features; * indicates a statistically significant difference between partner and recency adaptation ($p<.05$).

Table 3.13 shows the % of features that were adapted according to the adaptation

feature	Pr(+adapt)/Pr(Chance)		Adapt. Strength	
	partner	recency	partner	recency
Stem/POS	3.36	3.15	3.71	3.82
Stem/bigram	3.86	3.68	1.30	1.62*
Syntactic	3.09	3.36*	5.49*	4.99

Table 3.12: Average adaptation ratio and adaptation strength for *adapted* features; * indicates a statistically significant difference between partner and recency adaptation (p<.05).

	num	%adapted		avg. adapt. strength	
		partner	recency	partner	recency
p≥b	151.7	.14	.17	2.42	2.55
p≥b+1	78	.12	.14	3.47	3.59
p≥b+2	51.8	.12	.15	3.94	3.82

Table 3.13: Average distance measures for *adapted* features (Stem/POS only).[9]

measure and the corresponding adaptation strength for all adapted features. I considered three sets of features with varying frequencies in *prime* in the 32 *partner* and 32 *recency* priming dialog pairs. The first set includes all features where frequency in prime is at least equal to the baseline frequency ($p \geq b$); the second set includes all features where frequency in prime is at least one above the baseline ($p \geq b+1$); and the third set includes all features where frequency in prime is at least two above the baseline ($p \geq b+2$). *Num* column indicates average number of features examined in each feature set.

The results show that with the increase in *prime* frequency the % of adapted features does not increases for neither partner nor recency adaptation. This indicates that adaptation is not affected by frequency in prime. However, adaptation strength increases for both partner and recency adaptation. This indicates that for the adapted features, the feature frequency in the *target* is affected by the frequency in the *prime*.

I hypothesize that a speaker's lexical and syntactic choices in a dialog are affected by the speaker's dialog model. My results suggests that a speaker's conversation model is a combination of previous conversations by the speaker. The difference in the types of features adapted to the most recent dialog and to the dialog with the same partner suggests that the most recent and specific partner affect different parts of the speaker's dialog model. For example, the use of the first-person pronoun or negation is pervasive in the very next conversation with a different partner but not in the following conversation with the same partner, while the use of the direction-specifying prepositions, such as *across* and *through*, is pervasive in the dialogs with the same partner but not in the very next dialog with a different partner. This suggests that some sections of the speaker's model are affected by the most recent user's experiences, some sections are affected by the partner-specific experiences, and some are affected by both.

3.5 Discussion

In this chapter I presented two methods for measuring adaptation in dialog. The *adaptation ratio* measure, a variation on Church's measure of adaptation, evaluates how likely a feature is to appear in a target document with frequency greater than average if it appears in the prime document with frequency greater than average. The *adaptation strength* measure evaluates the strength of adaptation, the increase in frequency of adapted features. These measures have several advantages over those used in previous work. Comparing the frequency to an average instead of using a binary 'occurred'/'did not occur' distinction allows us to measure adaptation on both frequent and infrequent features. Comparing *adapt+* to *chance* allows us to measure adaptation in a relatively small corpus of dialog pairs. Evaluation of adaptation strength allows us to measure adaptation of a feature in single dialog pair (instead of across a whole corpus).

I used these measures to compare adaptation in partner- and recency-primed dialog pairs. I showed through a series of analysis of the Maptask corpus that these measures can identify features that exhibit adaptation and can be used across dialogs to evaluate the presence and strength of partner and recency adaptation.

I found that speakers exhibit adaption of both types. I outlined differences in the features that are adapted to recency and to the partner model: syntactic features tend to be adapted to the most recent partner. Lexical features with a personal pronoun '*I*' are adapted to the most recent partner. Features indicating direction (*across, through, about the*), and features with the pronoun '*you*' tend to be adapted to a specific partner, regardless of recency.

The findings of this work suggest that speakers adapt to spacial perspective and to the task-specific *strategy*, where the strategy is closely related to the speaker's lexical choices. Perhaps, in the case of adaptation to recency, if a speaker perceives that previously used strategy was successful, he/she will be likely to use the same strategy in the next conversation. In the case of adaptation to the partner, the speaker may remember which strategy the was used in their previous conversation and use the same strategy.

In this work I compared adaptation effect exhibited by different features and compared it between partner and recency. An interesting direction for future work is to determine the proportion of dialogs exhibiting adaptation and to investigate whether partner adaptation has an effect on quality of a conversation with a repeating partner, such as task success, dialog length, number of misunderstandings and clarifications. Reitter found that adaptation within a dialog positively correlates with task success. I would like to investigate whether partner-specific adaptation between dialogs has a similar effect on task success.

Chapter 4

Adaptation and System Building

Adaptation in a dialog system can be either *local* or *global*. In the *local* approach an existing system may be modified to support adaptation in one of its components. Local adaptation implementation is minimally invasive and may be independent of the overall system architecture. Examples of local adaptation include adaptation in speech recognition (Yu et al., 2000; Riccardi and Gorin, 2000; Soltau, 2005) and language generation (Walker et al., 2004; Stent et al., 2004) components. In the *global* adaptation approach, the system design supports adaptation. Multiple components may be involved. For example, Kempson et al. (2009) propose an adaptive dialog system with merged parsing and generation components.

In my work I do not redesign system architecture to incorporate adaptation. Instead, I use lightweight local adaptation and apply it to existing systems. In this chapter I describe three spoken interface systems that use different architectures: the Rate-A-Course dialog system developed at Stony Brook, the Let's Go! dialog system developed at CMU, and a spoken question answering interface developed partly at Stony Brook. I show how local adaptation is applied in each of these systems. I use the *Let's Go!* dialog system and the spoken question answering system for the experiments described in the following chapters.

4.1 *Rate-A-Course* Survey Dialog System

Surveys are a natural and commercially viable application for spoken dialog systems. Survey dialog systems also present interesting opportunities for research on spoken dialog and on survey design. The Stony Brook Rate-A-Course system is a survey dialog system that permits college students to evaluate their courses over the telephone. The Rate-A-Course system is a prototype telephone-based spoken dialog system that could be used as a replacement for or adjunct to other course evaluation methods. The novelty factor of talking to a dialog system might increase response rates; because the survey results (including comments in response to open-ended questions) are available in electronic form, they can be distributed over the web or telephone.

4.1.1 System Description

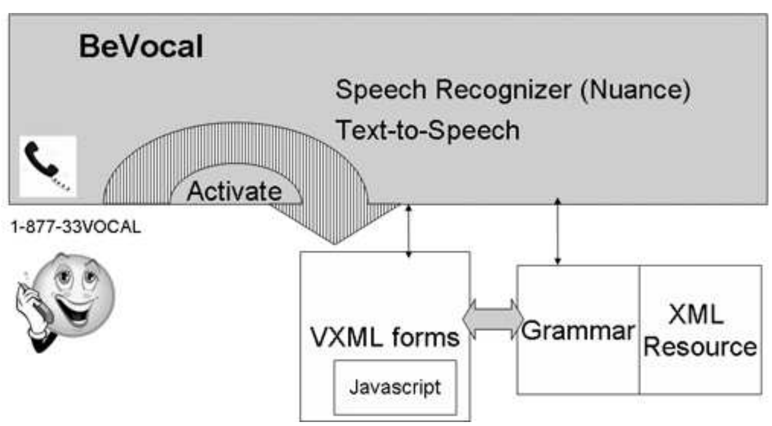

Figure 4.1: Rate-A-Course system architecture

The *Rate-A-Course* system is implemented in VoiceXML, XML and Javascript. It runs on the BeVocal Cafe platform and uses Nuance speech recognizer (BeVocal). Figure 4.1 shows the architecture of a system deployed on the BeVocal. A system developer provide VoiceXML forms, corresponding speech recognition grammars and resource files. Speech recognition, text-to-speech, and a phone line are provided by

BeVocal. When a user calls and enters an application code, the corresponding VXML form is activated and executed.

In the Rate-A-Course system the survey questions and potential answers are stored in the resource XML document that can be automatically generated from a web-based survey design interface. This interface permits the selection of choice points (for subdialogs), question types and error-handling strategies. The XML document is used to automatically generate speech recognition grammars and to populate VoiceXML forms that act as templates for different question types. Javascript embedded in the VoiceXML forms permits automatic logging of all system and respondent interactions.

Topic	**Synonyms**	**Answers/Ratings**
Instructor	teacher, professor	very good/100, good/75, okay/50, bad/25, very bad/0
Exams	tests, midterms	too hard/0, hard/50, about right/100, easy/50, too easy/0
Class size	course size, size of the class	too packed/0, packed/50, about right/100, small/50, too small/0
Assignments	homeworks, course work	too hard/0, hard/50, about right/100, easy/50, too easy/0
TA	t a, teachers assistant	very good/100, good/75, okay/50, bad/25, very bad/0

Table 4.1: Topics used in *Rate-A-Course* system experiment

Survey respondents are asked about five topics for a course that is being evaluated. For each topic, they are first asked to rate that aspect of the course (e.g. "Was the instructor very good, good, okay, bad or terrible?"). Then, they were asked to explain their rating (e.g. "Why did you think the instructor was okay?"). Table 4.1 gives information about the course topics.

Possible answers to closed-ended questions (e.g. "Was the instructor very good, good, okay, bad or terrible?") and question-related keywords taken from the XML document are used to create recognition grammars; these permit respondents to answer closed-ended survey questions using full or partial sentences, using the terms specified in the question or using synonyms of question terms. In this version of the system, no attempt is made to automatically process the answers to open-ended questions (e.g. "Why did you think the instructor was okay?") during the survey.

The *Rate-A-Course* system permits respondents to ask for the last question to be repeated and to ask for help at any time. A request for help is interpreted as a request for clarification of the current question. The system also provides help on a recognition failure or no input; this help can be a simple repetition of the question, an explanation of the answers or an example answer, or a subdialog, depending on the XML specification for the survey. Respondents in the experiment described here were allowed to go back or cancel only for certain questions (e.g. course department and number).

The *Rate-A-Course* system generates structured logs in the form of question-answer pairs for all questions, as well as a complete dialog history with pointers to the audio files containing respondents' speech.

System	S: We will now ask your opinion on the following aspects of your course: the instructor, the assignments and the exams. Is the instructor: very good, good, okay, bad, or very bad?
Mixed	S: Which topic was the next most important to you? Your choices are ...
User	S: Would you like to discuss another topic? U: Yes S: Which topic is the next most important to you? Your choices are ...

Table 4.2: System questions by initiative condition

The *Rate-A-Course* system implements several different dialog behaviors leading to different amounts of system interactivity:

- **Choice of question type:** In the XML document, survey designers can specify whether a question should be open-ended or closed-ended and can specify valid answers to a close-ended question.

- **Question ordering:** The VoiceXML forms implement random ordering of questions when the survey designer does not specify question order.

- **Initiative:** There are VoiceXML forms for a survey with *system initiative* (the system chooses the question order), *mixed initiative* (the respondent chooses in which order to answer survey questions), or *user initiative* (the respondent chooses which survey questions to answer, as well as the order in which to answer them). Table 4.2 shows how the system moves from topic to topic in each initiative condition.

The survey designer can give the respondent a code that pre-specifies system behaviors. This means that in addition to collecting survey data, the system can be used to perform research about spoken dialog and survey design.

4.1.2 Adaptation in *Rate-A-Course*

In this section I describe local adaptation in the *Rate-A-Course* system. This simple implementation allows us to study whether system adaptation has an effects on the user.

The Rate-A-Course system implements adaptation in lexical realization. It runs in one of the adaptation conditions (adaptive/non-adaptive). The *concepts* of the system are the topics that may be evaluated by a user: teacher, exams, homework, etc. Each concept has several synonimous forms of realization (e.g. teacher/lecturer/instructor). The system at first uses a randomly-selected form of each survey concept. In the adaptive condition the system switches after a user's utterance to the user's choice of concept realization (e.g. if a user says *"the instructor was very good"*, the system will then use the word *instructor* in the following prompts). In the non-adaptive condition the system never switches its initially chosen form of a concept. The system also implements syntactic adaptation. In the adaptive condition, the system switches its

id	S/U	Utterance
1	S:	Please choose one of the following: **instructor**, class size, text book, etc.
2	U:	**instructor**
3	S:	**Is** the **instructor** very good, good, okay, bad, or very bad
4	U:	very good
5	S:	Please describe why do you think **instructor** is very good
6	U:	the **professor was** very funny at times also entertaining as well as very informative and [um] it was obvious to all the students that it was a very intellectual person [uh] definitely knew what he was talking about and [um] yeah it was a good class
7	S:	Please choose one of the following: teaching assistant, class size, text book etc.
8	U:	**T. A.**
9	S:	**Was** the **T. A.** very good, good, okay, bad, or very bad
10	U:	very good

Table 4.3: An example dialog with the Rate-A-Course system where the system adapts to the user

tense (the form of the verb is/was) to the tense used by the user.

In the example dialog shown in Table 4.3, the user first chooses to discuss the instructor and then the T.A. *T.A.* and *teaching assistant* are the two alternative realizations of the same concept. In the question (7), the system used *teaching assistant*. The user chooses to discuss the *T.A* in (8). The system switches its vocabulary and uses the word "T.A" in the next utterance (9). In the free response (6), the user uses past tense *"professor was"*. In the following utterance (9), the system switches to using past tense as well.

It has been shown that adaptation in human-human conversations correlates with rapport between speakers. Comparison of user's answers in communication with adaptive system and non-adaptive systems allows to study user's perception of a dialog system and determine whether users of adaptive system be more sincere in their discussion. Unfortunately, in our study we did not have enough participants to make conclusions about adaptation.

4.2 *Let's Go!* System: Online Bus Information

4.2.1 System Description

The *Let's Go!* Raux et al. (2005) system is developed, maintained, and deployed at Carnegie Mellon University. This telephone-based system provides information about bus routes, departure times, and bus connections in Pittsburgh. The system is reachable through the local Port Authority number outside of the hours when human operators answer the phone lines. The system receives calls from users of all ages and language backgrounds.

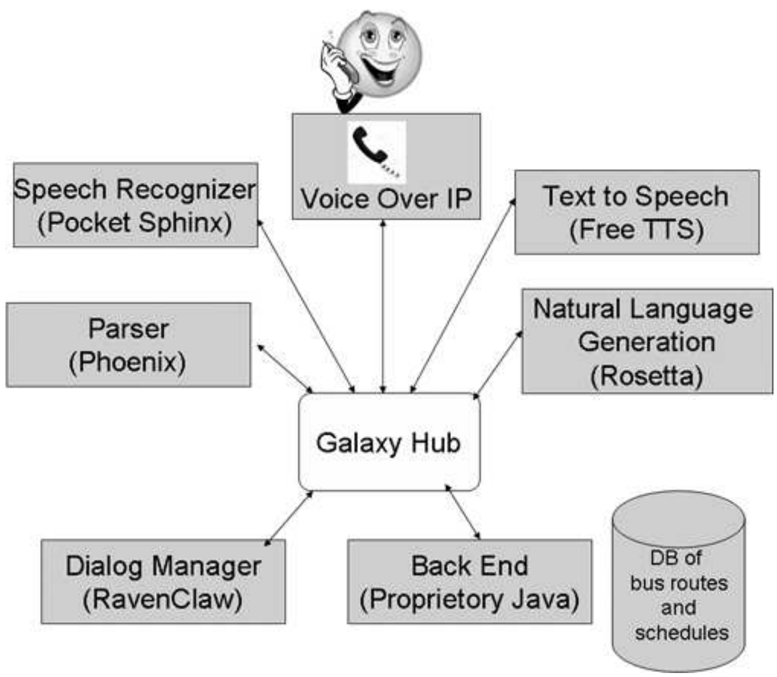

Figure 4.2: *Let's Go!* system architecture

Let's Go! was developed using the distributed Olympus framework with an architecture shown on Figure 4.2. System components run as separate applications communicating with messages through central hub. The system uses the Pocket Sphinx speech recognizer (Huggins-Daines et al., 2006). The result of the recognition is parsed by Phoenix, a robust parser using an extended context-free grammar. It allows the system to skip unknown words and perform partial parsing (Ward and Issar, 1994a). The dialog manager is developed using RaveClaw (Bohus and Rudnicky, 2003) that allows "object-oriented" specification of a dialog structure. The dialog in RavenClaw architecture is defined as a graph. Each node in the graph is a minimal dialog component that performs a specific action and has pre- and post-conditions. The dialog flow is determined by edges between nodes. With this structure, the reuse of minimal dialog components inside the dialog manager is maximized. The Rosetta template-based generator is used for response generation. The speech synthesis component is the off-the shelf open source Free TTS system. Other research Olympus-based dialog systems developed at CMU include the Room Line, LARRI (Bohus and Rudnicky, 2002), and Calendar (Stenchikova et al., 2007)[1] systems.

To accommodate the diverse user population and noisy speaking conditions the dialog system is designed as a linear system-initiative dialog. The system asks the user for several concepts sequentially: an optional bus route number, a departure place, a destination and a desired travel time. Each concept is explicitly confirmed. The backend of the system is the database of all stops and bus routs in Pittsburgh provided by the Port Authority. Table 4.4 shows a sample dialog with the system.

The system receives on average 40 calls per day. Average call length is 12.9 turns with a large standard deviation. A 2005 call analysis showed a speech recognition error rate of 68%. The task success rate is estimated at 43%.

[1] The first dialog systems based on the Ravenclaw/Olympus framework to be developed outside of CMU.

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 4.4: Sample dialog with *Let's Go!*

Researchers at Carnegie Melon transcribed and hand-annotated the first two months of *Let's Go!* system operation in 2005 (2411 dialogs), and one month in 2006 (1430 dialogs). Since 2007, CMU researchers have provided the system as a resource for outside developers. It is a valuable resource for the dialog system research community as it provides a testbed on a real system for experimenters and a large and stable pool of users. I was granted permission to run two adaptation experiments on the *Let's Go!* system. These experiments are described in Chapters 5 and 6.

4.2.2 Adaptation in *Let's Go!*

I use the system for *directive* and *responsive short-term* adaptation experiments with real users. I modify the natural language generation component to use different phrasing of the prompts. This allows me to evaluate immediate effect of the system prompt on a user. In Chapter 5 I describe my experiments evaluating whether users are likely to use the same syntactic structures and lexical choices as the system. I compare user responses to the departure location prompt with different phrasing, (e.g. *Where are you leaving from?* and *What is the place of your departure?*).

These experiments are achieved by the local modification of natural language component of the system, Rosetta. I create four natural language generation conditions. The conditions differ between each other by presence of verbs and prepositions. I parametrize Rosetta, such that an input parameter defines which of the four types of four conditions is used. I modify all of the system prompts for each condition.

Users' Lexical and syntactic choices may be affected by the system throughout the dialog. In my directive adaptation experiment I am interested in user's syntactic and lexical choices in response to the initial system prompt and not further on in the dialog.

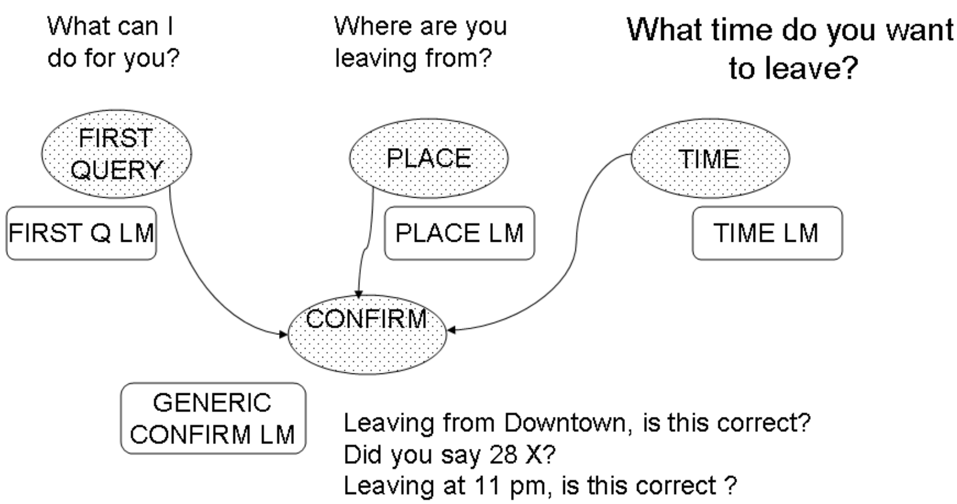

Figure 4.3: Dialog states and language models used in *Let's Go!*

Let's Go! system implements local adaptation in its speech recognition. Speech recognition is a statistical process that uses two kinds of models: an acoustic model and a language model. An acoustic model is generated from spoken data with aligned transcription. Acoustic model records the probabilities of mapping acoustic frequency features to lexical units. A language model records probabilities of n-grams (strings of length n) occurring in an utterance. The similarity between the recognized user utterances and the dataset used for generating a language model affects recognition performance.

In order to provide the user with route information, *Let's Go!* elicits a departure location, a destination, a departure time, and optionally a bus route number. *Let's Go!* has four dialog states corresponding to the information it elicits: *first-query*, *place*, *time*, and *confirm*. Figure 4.3 illustrates the dialog states currently used in *Let's Go!*.[2] Initially the system is in the *first-query* dialog state, in which it asks a general question *What can I do for you?*. *Let's Go!* is a flexible input system. It allows users to specify any combination of concepts in each state. For example, as an answer to *first query* a user can specify all of the information about the route, e.g. *Going from Downtown to Oakland at four p.m.*, or only part of the information, e.g. *Leaving from Downtown.* To answer the place prompt, *Where are you leaving from?*, users are likely to specify a place concept, however they can also take task initiative and specify other concepts. Each concept value provided by the user is explicitly confirmed by the system.

In each of the system states, a state-specific language model is used for recognizing a user's answer. The state-specific language models are trained on user utterances from the corresponding system states from previously annotated user interactions with the system. The system's speech recognizer *adapts* to the context of a dialog as it switches language model used for recognition. In Chapter 6 I describe my experiments with further adaptation of a system's language model for content of user utterances in *confirm* system state.

[2] There is also *next-query* which is similar to *first-query* and is omitted from the diagram

4.3 Spoken Interface for Question Answering

Question answering (QA) is the task of automatic retrieval of an answer given a question (e.g. *Who invented silly putty?* or *When was Mozart born?*). Question answering provides a natural language interface for information retrieval. This interface also opens the possibility of access to information retrieval using voice. Prior to answer retrieval, a question has to be recognized. Spoken interface for QA recognizes the spoken question and passes it to QA system to retrieve an answer.

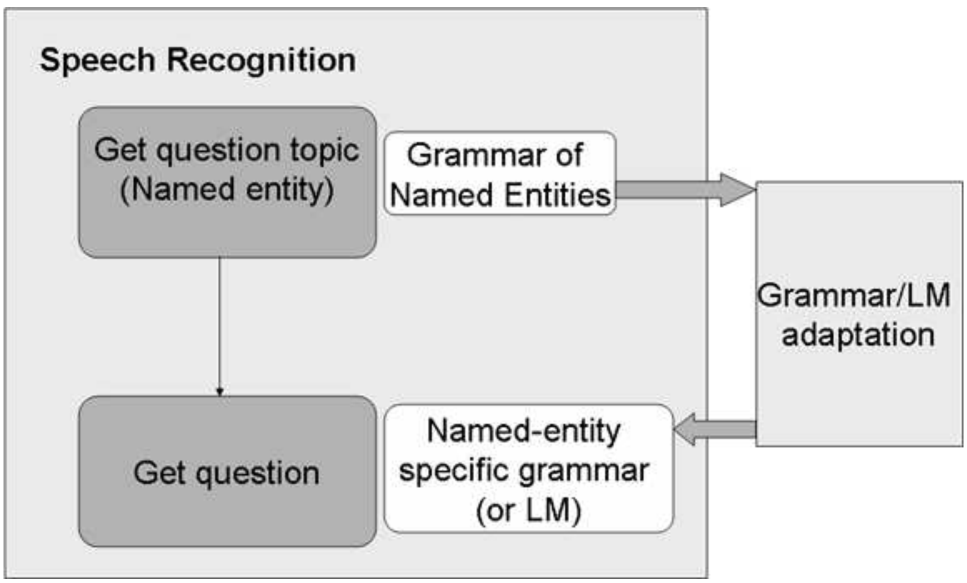

Figure 4.4: Spoken question answering system architecture

Figure 4.4 shows a diagram of an adaptive spoken QA interface. The proposed interface first asks the user to specify a question topic. For example, a topic of *When was Mozart born* is a named entity *Mozart*. This information is used to create a grammar or a language model for the recognition of the question. Next, the question is recognized using the topic-specific language model.

In my experiments I evaluate the potential performance improvement of the speech recognition on questions with the proposed adaptive system architecture.The results of the experiment are described in Chapter 7. In the future work I would like to implement the proposed architecture and address the performance of named entity recognition.

Adaptation in the speech recognition of spoken question answering is similar to the adaptation in speech recognition of a dialog system described earlier in this chapter. In both cases the language model used for recognition of a user's utterance is adapted according to the system's expectation about the user's utterance. In the *Let's Go!* dialog system, the language model is adapted to the expectation of the concept that the user specifies (place, bus, or time). In the spoken interface to QA, thelanguage model is adapted to the topic (or named entity) of the question. From the system building perspective, both systems have an additional component (a classifier in *Let's Go* and a dynamic language model builder in spoken QA) that make a selection of the model.

Chapter 5

Directive Adaptation in Dialog

5.1 Motivation and Research Goals

In this chapter I describe my experiments on *directive adaptation*. The goal of these experiments is to explore user behavior in response to varying system conditions. In this study I investigate whether dialog system users in noisy real-world conditions adapt to system prompts as they do in a conversation with another person or with a dialog system in controlled experimental conditions. This is the first study, to the best of my knowledge, that investigates the adaptive behavior of real users of a live dialog system. Previous research on user adaptation to dialog systems was conducted in laboratory settings (see Section 2.1). However, the behavior of recruited subjects in a quiet laboratory may differ from that of real users in the noisy world because of the user's real needs and eagerness to complete the task. Experimental users may have a different strategy of dealing with system errors in a dialog. For example, users of an airline reservation system (Walker et al., 2002) in experimental settings often switch their departure or destination location after continuous misunderstandings. I hypothesize that a user with a real need to depart from the particular location would use a different strategy to overcome system's recognition error and be more eager to complete the task (Ai et al., 2007). External stimuli, such as noise and varying quality of cell-phone signal, may also affect the user's interaction with the system.

In the experiments described in this chapter analyze dialogs from the CMU's *Let's Go!* dialog system (Raux et al., 2005). The results presented here confirm prior results showing that users exhibit adaptation to the system's lexical and syntactic choices. I observe statistically significant differences in users' lexical and syntactic choices between system conditions as I vary the form of system prompts. I observe statistically significant differences in the system's ability to detect task-related concepts in user utterances as I vary the form of system prompts. In the following sections I describe my analysis of: effect of the dialog system's lexical and syntactic choices on user responses (Section 5.2), and effect of the system's choice of concept form on the user's choices of concept form (Section 5.3).

5.2 Lexical and Syntactic Variation in System Queries

In this experiment I analyze user adaptation to the presence of a verb/preposition and to the form of a function verb in the system's prompt. I also analyze the effect of the system's lexical and syntactic choices on dialog performance.

Knowledge about user adaptation to the words or syntactic structures in the system's prompt can be particularly useful in *flexible input* dialog systems. *Flexible input* dialog systems allow the user to respond to system prompts with phrases and sentences and specify information other than that currently requested. *Flexible input* systems may also allow the user to take task initiative. *Limited input* dialog systems, on the other hand, require the user to respond to each system prompt using only the concept and words currently requested by the system. Speech recognition (ASR) accuracy in *limited input* systems is better than in *flexible input* systems (Danieli and Gerbino, 1995; Smith and Gordon, 1997). However, depending on a task, *flexible input* systems may achieve better overall system performance (Chu-Carroll and Nickerson, 2000; Smith and Gordon, 1997). With user adaptation, in *flexible input* dialog systems prompts can be formulated to maximize ASR accuracy and reduce the number of ASR timeouts (Sheeder and Balogh, 2003).

5.2.1 Experimental Design

Speaker	Task type	Utterance
Sys	Open	Welcome to the CMU Let's Go bus information system. What can I do for you?
Usr		*61A schedule*
Sys	Request Departure Location	Where do you wanna leave from?
Usr		*From downtown*
Sys	Confirm Departure Location	Leaving from downtown. Is this correct?
Usr		*Yes*
Sys	Request Arrival Location	Where are you going to?
Usr		*Oakland*
Sys	Confirm Arrival Location	Going to Waterfront. Is this correct?
Usr		*No, to Oakland*

Table 5.1: Sample dialog from *Let's Go!* with labeled system task type

I conducted my experiment using the *Let's Go!* telephone-based spoken dialog system that provides information about bus routes and is described in Section 4.2. The users are naive callers seeking information about buses. In order to provide the user with route information, *Let's Go!* elicits a departure location, a destination, a departure time, and optionally a bus route number. *Let's Go!* is a *flexible input* dialog system. The user can respond to a system prompt using a single word or short phrase, e.g. *Downtown*, or a complete sentence, e.g. *I am leaving from downtown*[1]. Each concept value provided by the user is explicitly confirmed by the system. Table 5.1 illustrates an example dialog with the system.

The variables in this experiment are 1) presence of a verb and/or prepositions in a system prompt; and 2) a verb form in a system prompt. I chose to focus on the verbs *leaving/leave* and *going/go* because according to the preliminary corpus analysis these are the most frequently used verbs by the users of the *Let's Go!*. In this experiment I had enough time to run four experimental conditions. Hence, I decided to focus the experiment on adaptation to the presence of a verb/preposition and adaptation to the verb form. In future work I would like to also include the verbs *departing* and *arriving* in the experimental variables and expand the study by measuring adaptation to the verb choice (*leaving* vs. *departing*) as well as the perspective choice (*going to* vs. *arriving at*).

cond	request departure location	confirm departure location	request arrival location	confirm arrival location
1	Where are you **leaving from**?	**Leaving from X**, is this correct?	Where are you **going to**?	**Going to** X, is this correct
2	Where are you **leaving from**?	**From** X, is this correct?	Where are you **going to**?	**To** X, is this correct
3	What is the place of your departure	X, is this correct?	What is the place of your arrival?	X, is this correct
4	Where do you want to **leave from**?	You want to **leave from** X, is this correct?	Where do you want to **go to**?	You want to **go to X**, is this correct

Table 5.2: Experimental conditions

I ran four experimental conditions, varying the lexical choices and syntax of system

[1]The user response can also contain concepts not requested in the prompt, e.g. specifying departure location and bus number in one response.

prompts for the *request departure location*, *request arrival location*, *confirm departure location*, and *confirm arrival location* tasks (see Table 5.2). System prompts in each system condition differ by presence of a verb (*to leave, to go*) or a preposition (*to, from*), and by the syntactic form of the verb. The *request location* prompt contains a verb in only two of the experimental conditions (1 and 4). The *confirm location* prompt contains both a verb and a preposition in conditions 1 and 4, only a preposition in condition 2, and neither a verb nor a preposition in condition 3. In conditions 1 and 4, both request and confirmation prompts differ in the verb form (*leaving/leave, going/go*).

The practical motivation for this experiment is a potential improvement of a speech recognition in the system. I hypothesize that knowledge about a user's utterance may help improve speech recognition performance in a dialog system. This study aims to determine whether lexical and syntactic choices in a system prompt help predict content of a user's utterance. In case of a correlation between presence of a verb and its form in a system prompt and in a user's utterance, the system can 1) guide users into using verbs, prepositions, or particular verb forms; and 2) dynamically adapt ASR component based on the expected content of a user's utterance.

5.2.2 Experimental Data

I collected 2184 dialogs (over 500 for each experimental condition).

In the Sections 5.2.3 and 5.2.5, I describe analysis done using automatically recognized user utterances on the whole dataset of 2184 dialogs. Although the data contains recognition errors, the only difference in system functionality between the conditions is the formulation of the system prompt.

In order to confirm my conclusions from my analysis of automatically recognized utterances, I manually transcribed a subset of 143 dialogs where the speech recignizer recognized a verb. The purpose of this exercise was to 1) transcribe data for a contribution to the *Let's Go* Project and 2) confirm conclusions on lexical and syntactic adaptation from my analysis of recognized output. The results on the transcribed dataset are reported in Section 5.2.4.

5.2.3 Results: User Adaptation to System Lexical Choice

I analyze whether users are more likely to use action verbs (*leave, leaving, go*, or *going*) and prepositions (*to, from*) in response to system prompts that use a verb or a preposition. This analysis is interesting because automatic speech recognition partially relies on *context words*, words related to a particular concept type such as place, time or bus route. For example, the likelihood of correctly recognizing the location *Oakland* in the utterance *"going to Oakland"* is different from the likelihood of correctly recognizing the single word utterance *"Oakland"*.

Cond.	Sys uses verb	Sys uses prep	% with verb	% with prep
Users' responses to *request location* prompt				
(1)	yes	yes	2.3% ∗	5.6%
(2)	yes	yes	1.9%	4.3%
(3)	**no**	**no**	**0.7%**	4.5%
(4)	yes	yes	2.4%∗	6.0%
Users' responses to *confirm location* prompt				
(1)	yes	yes	15.7% ∗ ♠	23.4%
(2)	**no**	yes	**3.9%**	16.9%
(3)	**no**	**no**	**6.4%**	12.7%
(4)	yes	yes	10.8%	22.0%

Table 5.3: Percentages of utterances containing verbs and prepositions. ∗ indicates a statistically significant difference ($p<.01$ with Bonferroni adjustment) from the *no action verb* condition highlighted in **bold**. ♠ indicates a statistically significant difference from the *no action verb in confirmation* condition (2).

Table 5.3 shows the percentages of user responses in each experimental condition that contain a verb and/or a preposition. I observe adaptation to the presence of a verb in the system prompt in user responses to *request location* prompts. The prompts in conditions 1, 2 and 4 contain a verb, while those in condition 3 do not. The differences between conditions 1 and 3, and between conditions 4 and 3, are statistically significant ($p<0.01$)[2]. The difference between conditions 2 and 3 is not statistically significant, perhaps due to the absence of the verb in the *confirm location* prompt giving less priming.

A similar adaptation to the presence of a verb in the system prompt is seen in user responses to *confirm location* prompts. The prompts in conditions 1 and 4 contain a verb while those in conditions 2 and 3 do not. The differences between conditions 1 and 2, and between conditions 1 and 3, are statistically significant ($p<.01$), while the difference between conditions 2 and 4 exhibits a trend. I hypothesize that the lack of the statistically significant differences between conditions 2 and 4, and conditions 3 and 4, is caused by the low relative frequency in the dataset of dialogs in condition 4.

I do not find statistically significant differences in the use of prepositions. However, I observe a trend showing higher likelihood of a preposition in user responses to *confirm location* in the conditions where the system uses a preposition. Prepositions are short closed-class context words that are more likely to be misrecognized (Goldwater et al., 2008). Hence, more data (or human transcription) may be required to see a statistically significant effect. More detailed analysis of prepositions is part of the future work for this project.

5.2.4 Results: User Adaptation to System Verb Form

I analyze whether the system's choice of a particular verb form affects the user's choice of verb form. For this analysis I only consider user utterances in response to a *request location* or *confirm location* prompts that have an automatically identified

[2]All analysis in this section are t-tests with Bonferroni adjustment.

Condition	Usr: LEAVING	Usr: LEAVE	total
Progressive system form	**74.5%**	25.5%	55
Simple system form	43%	**57%**	42
Neutral system form (unprimed)	61.3%	38.7%	31
Condition	GOING	GO	total
Progressive system form	**84.4%**	15.6%	45
Simple system form	46.5%	**53.5%**	43
Neutral system form (unprimed)	66.6%	33.4%	21

Table 5.4: Usage of verb forms in user automatically recognized utterances

concept and contain at least one of the verb forms *leaving, going, leave*, and *go*[3].

Table 5.4 shows the total counts and percentages of each verb form in the *simple form priming* condition (condition 4), the *progressive form priming* condition (condition 1), and the *neutral* condition (condition 3)[4]. I find that the system's choice of verb form has a statistically significant impact on the user's choice (χ^2 test, p<0.01). In the *neutral* (unprimed) condition, users are more likely to choose the progressive verb form. In the *progressive form priming* condition, this preference increases by 13.2% for the verb *to leave*, and by 17.8% for the verb *to go*. By contrast, in the *simple form priming* condition, this preference decreases by 18.3% for the verb *to leave* and by 20.1% for the verb *to go*, making users slightly more likely to choose the simple verb form than the progressive verb form.

I manually transcribed 63 dialogs in the *simple form* priming condition and 80 dialogs in the *progressive form* priming condition. To maximize the number of dialogs of interest (that contain a verb in a specification of a place) I used automatic speech recognition to guide the selection of dialogs for transcription. I selected the dialogs where the automatic speech recognizer recognizes any of the words *go*, *going*, *leave*, *leaving* anywhere in the dialog.

[3]Such utterances constitute 3% of all user responses to all *request* and *confirm place* prompts in the dataset.

[4]I ignore condition 2 where the verb is used only in the *request* prompt.

Table 5.5 shows the percentages of each verb form in the transcribed data. These are proportions of utterances that contain a *place* concept and occur anywhere in the dialog. Similarly to the result on recognized speech, this result shows a strong indication that users are more likely to use the same verb form as the system (χ^2 test, $p<0.01$).

Condition/user's verb	Usr: LEAVING	Usr: LEAVE	total
Progressive system form	90% (45)	10% (5)	50
Simple system form	40% (12)	60% (18)	30
Condition/user's verb	Usr: GOING	Usr: GO	total
Progressive system form	88% (60)	12%(8)	68
Simple system form	37% (16)	63% (27)	43

Table 5.5: Usage of verb forms in users' transcribed utterance

5.2.5 Results: Prompt Design and System Concept Detection

The correct identification and recognition of task-related concepts in user utterances is an essential functionality of a dialog system. Table 5.6 shows the percentage of user utterances following a *request location* prompt that contain an automatically-recognized location concept. Note that this analysis is of automatic concept identification and is performed on the speech recognition output of 2184 dialogs. Automatic concept identification does not directly correspond to recognition accuracy, but on a large dataset it approximates the recognition accuracy. Condition 4, where the system prompt uses the verb form *to leave*, achieves the highest concept identification rates. The differences in concept identification rates between conditions 1 and 4, and between conditions 3 and 4, are statistically significant for *request arrival location* (inference on proportions test, $p<.01$). Other differences are not statistically significant, perhaps due to lack of data.

System prompt	Arrival request	Departure request
(1)	72.2% $*$	63.8%
(2)	77.4%	61.0%
(3)	74.5% $*$	61.5%
(4)	**82.0%**	66.0%

Table 5.6: Concept identification rates following *request location* prompts

5.3 Concept Form Variation

Concepts in a human-computer dialog convey task-specific information to the system. Concept recognition is essential for the system's ability to handle the task. In *Let's Go!* system concepts are names of neighborhoods (*Downtown, Squirrel Hill,* etc.), bus routes (*28X, 61A* etc.), and time (*now, four a.m., seven o'clock,* etc.). In an airline system, concepts would also include departure and arrival cities, airport names, and dates. Concepts differ from other words in a user's utterance because they contain task-required information.

In this work, I investigate whether users adapt to the form of a concept used in the system's prompt. In addition to providing further evidence of convergence in human-computer dialog, the findings of this experiment have implications for dialog system design. Currently, much dialog systems research is devoted to improving ASR accuracy, because this is a significant contributor to task success rates and to dialog length. If users adapt to the systems choices of realization for task-related concepts, we can predict the users choice of realization and use this to adjust the systems language model, improving ASR accuracy specifically on concept words. Another way to improve ASR accuracy is to guide the user into using words that are likely to be recognized correctly (Hockey et al., 2003; Sheeder and Balogh, 2003; Tomko and Rosenfeld, 2006). In Chapters 6 and 7 I describe speech recognition experiments where prediction of a concept in an utterance improves recognition of the utterance.

I hypothesize that prediction of concept form can similarly lead to speech recognition improvements.

Study of adaptation in *concept form* may have implication for both *limited* and *flexible* input dialog systems. To complete a task, users of any task-oriented dialog system must specify system-required concepts. When a concept has multiple realization forms (e.g. *four p.m.* and *four in the afternoon*), users must choose one of the forms to use in an utterance. When users of a *flexible input* dialog system specify time, they can say a full sentence (e.g. *I am leaving at four*), a phrase (e.g. *at four*), or simply a concept (e.g. *four*). When users of a *flexible input* dialog system specify destination, they can use different syntax (e.g. *going to Downtown*, *arriving Downtown*), or, again, use a single concept (e.g. *Downtown*). While the non-concept words in a user's utterance (e.g. *I, am, leaving, at, arriving, etc.*) are optional, presence of a concept (e.g. *four, Downtown*) is essential. The user of a *limited input* dialog system has less flexibility in the choice of non-concept words than the user of a *flexible input* dialog system. However, the user of a *limited input* dialog system has the same flexibility in the choice of concept form as the user of a *flexible input* system.

In this work I investigate adaptation to the *time* concept because *time* has multiple different realizations. To indicate the same time, a user may say *four*, *four o'clock*, *four p.m.*, or *four in the afternoon*. All of these realizations of *time* are common English phrases and I can safely assume that users are familiar with each of the realizations. The correct recognition of the part of day specification in time concept is important in the *Let's Go!* system because the system makes an assumption about the part of day based on the time of a call. So, if a user calls at night to check the morning busses, the recognition of the part-of-day is essential. Table 5.7 shows the *time forms* used by users of the *Let's Go!* system and their relative frequencies in a *Let's Go!* corpus. I chose to study the *time* concept because it has the most variability in a *Let's Go!* dialog system. This variability is not unique to *time* and

time form	example realization	frequency
TIME_ONLY	four, five, six thirty...	31.1%
TIME_APM	four a.m., ten p.m., one fifteen p. m.	43.5%
TIME_POD	four in the morning/evening/afternoon/	4.6%
TIME_OCLOCK	five o'clock	16%
OTHER	four o'clock p. m., a. m.	4.8%

Table 5.7: Formats of the time in users' utterances and their relative frequencies in one month of *Let's Go!* 2006 dataset.[6]

the findings of this work may apply to other system concepts. *Place* names can have multiple realizations. For example, *SAC* and *Student Activity Center* refer to the same location at Stony Brook Campus. *Fifth and Madison* and *Madison and Fifth* refer to the same intersection in New York City.

I hypothesize that the user's choice of the concept form (*time form* in *Let's Go!*) is affected by the system's choice of the concept form. In the next sections I describe my experiment and results supporting this hypothesis.

5.3.1 Experimental Design

For this experiment, I use the *Let's Go!* dialog system described in Chapter 4.2.

I evaluate three time forms of system priming: TIME_ONLY, TIME_APM, and TIME_POD[7] (see Table 5.7 for examples). These time forms have different properties: TIME_ONLY is the most frequent form in the *Lets Go!* corpus, but it is potentially ambiguous as it can mean either night or day. The TIME_APM form is the shortest unambiguous form. TIME_POD is the long unambiguous form and has a very low frequency in the *Lets Go!* corpus. I chose to investigate TIME_ONLY and TIME_POD forms because they are the most frequently used forms by the system users according to the previously transcribed data. Another frequent time form is TIME_OCLOCK.

[7]POD stands for Pard-Of-Day

To keep the experiment tractable I did not investigate it in this experiment.[8] I chose to investigate TIME_POD because I was interested to test adaptation to an infrequently used time form.

I investigate the **directive** effect of the system prompts: whether users are likely to use the same form of the time as the system. I measure and compare the frequencies of each *time form* in user utterances in different experimental conditions. The **directive** effect of a system prompt on the user's form of concept would suggest that the system prompts have **directive power** to guide users into using concept forms that are easier for the system to automatically recognize. Ability to predict the form of a concept in a user's utterance allows grammar or language model adaptation of the ASR and NLU components to the expected concept form that can lead to improvement in speech recognition performance. I hypothesize that in dialog systems with more complex domains, such as tutoring or technical assistance, where more concepts may have diverse synonymous realizations, system's ability to guide a user into using a specific concept form may play an important role for speech recognition improvement.

In this experiment the system *primes* the user for one of the *time forms* in the prompt asking the user about departure time. I assume that *priming* of the *time* concept occurs when the system specifies the time. If the system uses the TIME_ONLY form, (e.g. *Are you leaving at four?*), I assume that the user was primed for the TIME_ONLY form. If the system uses the TIME_APM form, (e.g. *Are you leaving at four p.m.?*), I assume that the user was primed for the TIME_APM form. If the system uses the TIME_ONLY form, (e.g. *Are you leaving at four?*), I assume that the user was primed for the TIME_ONLY form.

In a normal conversation with a system, users specify *time* before the system can prime them (see Table 5.8). Only in a confirmation utterance, after the user

[8] It would have been desirable to study TIME_OCLOCK. I expect to observe similar results on the TIME_OCLOCK as on the TIME_APM.

id	Speaker	Utterance
1	Sys	What time do you want to leave?
2	Usr	at seven
3	ASR	at seven
4	Sys	Leaving at seven in the morning
5	Usr	yes

Table 5.8: Normal dialog flow with time request

1 - 7, 2 - 10, 3 - 6, 4 - 7, 5 - 9, 6 - 3,7 - 11, 8 - 10, 9 - 1, 10 - 1, 11 - 7

Table 5.9: Simulated error in time recognition

specifies time, the system says a time. However, if the time was correctly recognized by the system, the user does not repeat the time after the system's confirmation prompt. I use a trick to cause users to say a time after the system's confirmation. For experimental purposes, I made a modification to the system that allows users to be primed. After the initial time query: *What time are you leaving?*, the system simulates a recognition error. To make the system's error more realistic, the time in the simulated error is a time that is phonetically closer to the time (hour and minute) initially recognized by the ASR. The system's choice for the incorrect time to present to the user in a confirmation with a simulated error is shown in Table 5.9. A sample dialog with a simulated error is shown in the Table 5.10. In response to the simulated error, the user makes one of the four responses: 1) correct the system immediately, 2) answer negatively to the confirmation and then correct the system, 3) start a new query, and 4) hang up. In this experiment I am interested in user responses of type 1 and 2 as illustrated in Table 5.9.

Another possible method of priming the user would be through a forced help message. For example, after the system's time prompt *What time would you like to depart?*, the system could play an explicit help message: *For example, you can say four p.m.* Help messages in the *Let's Go!* system are played for the user when the

id	Speaker	Utterance
1	Sys	What time do you want to leave?
2	Usr	at seven
3	ASR	at seven
4	Sys	Leaving at one in the morning
	User response type 1	
1.5	Usr	no at seven a. m.
	User response type 2	
2.5	Usr	no
2.6	Sys	What time do you want to leave?
2.7	Usr	at seven a. m.
	User response type 3	
3.5	Usr	new query
	User response type 4	
3.5	Usr	HANGUP

Table 5.10: Simulated error in a dialog flow with time request

system perceives a communication problem. Under normal circumstances it is not desirable to burden users with excessive help. Also, the priming effect in an explicit help message may be stronger than priming in a normal conversation. To compare my results with Brennan (1996)'s, I choose the simulated error priming method for this experiment.

5.3.2 Experimental Data

System Condition	System's Confirmation Question	dialogs transcribed
SYS_TIME_ONLY	Leaving at T?	44
SYS_TIME_APM	Leaving at T a. m./p. m. ?	49
SYS_TIME_POD	Leaving at T in the morning/evening/afternoon ?	47

Table 5.11: Confirmation prompt and the number of dialogs transcribed for each system condition (T can be hour or hour+minutes)

User Utterances	Description	# utterances
Unprimed	Initial utterance before priming	141
First primed	First utterance after a simulated error prime	130
All primed	All utterances after a simulated error prime	258

Table 5.12: Number of *Unprimed*, *First primed*, and *All primed* utterances

I collected over 2000 dialogs with *Let's Go!* using this setup.[9] However, not all of these dialogs contained mention of the *time* concept. The most common user response to the *Let's Go!* system's prompt *When do you want to leave?* is *now*. The dialogs where users say *now* instead of explicitly specifying time are not useful for this experiment. I used automatic speech recognition output to guide me in selecting dialogs with user responses of type 1 and 2 in Table 5.10 by selecting dialogs where a time was recognized at least twice. I manually transcribed 50 dialogs for each condition.

Table 5.11 shows the system's prompts and number of transcribed dialogs for the three system conditions: SYS_TIME_ONLY, SYS_TIME_APM, and SYS_TIME_POD. The number of dialogs for each condition excludes the transcribed dialogs that did not contain mentions of time after a confirmation.

5.3.3 Results: User Adaptation to System Concept Form

If the user adapts to the systems time form, then we would expect to see a greater proportion of the systems time form in user utterances following the prime. I compare the proportion of three time forms (U_TIME_APM, U_TIME_ONLY, and U_TIME_POD)[10] in each system condition for 1) *unprimed*, 2) *First primed*, and 3) *All primed* user's

[9]The same set of dialogs was collected in the experiment described in Section 5.2.

[10]Corresponding to the time forms in Table 5.7. Prefix 'U' stands for 'user'.

utterances (see Table 5.12). *Unprimed* utterances are the user's initial specifications of time before the system's confirmation prompt. *First primed* utterances are user utterances immediately following system's confirmation (utterances 2.7 or 3.5 in Table 5.10). The *First primed* utterances are guaranteed to follow the system's confirmation prompt immediately. *All primed* utterances are all user utterances in a dialog following system's confirmation with the simulated error. *All primed* utterances include *First primed*, plus all consecutive utterances with time in the dialog (excluding utterances after the user says *new query*). The separation of *First primed* and *All primed* was inspired by Brennan (1996) who evaluated convergence on *immediate* (immediately following the priming prompt) and *delayed* (following later in dialog) user utterances. I measure the priming effect in *First* and *All primed* utterances separately because *All primed* utterances may not immediately follow the system's priming utterance and priming decays over time (Reitter et al., 2006a).

I hypothesize that for the *unprimed* user utterances there will be no difference in proportions of each of the examined time form among different system conditions. I hypothesize that for the *primed* (both *All* and *First*) user utterances each time form will be most frequent in the corresponding system condition than in the other two conditions. I predict that in *primed* user utterances 1) U_TIME_APM is more frequent in SYS_TIME_APM than in SYS_TIME_ONLY and SYS_TIME_POD; 2) U_TIME_ONLY is more frequent in SYS_TIME_ONLY than in SYS_TIME_APM and SYS_TIME_POD; and 3) U_TIME_POD is more frequent in SYS_TIME_POD than in SYS_TIME_APM and SYS_TIME_ONLY. Table 5.13 shows the proportions of each time form among all user utterances specifying time in each of the system conditions. To test statistical significance of the results I perform inference on proportions for a large sample.

U_TIME_APM

As expected, There are no statistically significant differences in the proportions of

Unprimed				
system/user	U_TIME_APM	U_TIME_ONLY	U_TIME_POD	U_OTHER
SYS_TIME_APM	25%	42%	8%	25%
SYS_TIME_ONLY	30%	52%	2%	16%
SYS_TIME_POD	24%	49%	4%	23%
First Primed				
system/user	U_TIME_APM	U_TIME_ONLY	U_TIME_POD	U_OTHER
SYS_TIME_APM	**49%**	29% ♠	2%	20%
SYS_TIME_ONLY	21% ♣	**58%**	0%	21%
SYS_TIME_POD	29%	45%	5%	21%
ALL Primed				
system/user	U_TIME_APM	U_TIME_ONLY	U_TIME_POD	U_OTHER
SYS_TIME_APM	**63%**	19% ♣	3%	15%
SYS_TIME_ONLY	21% ♣	**50%**	2%	27%
SYS_TIME_POD	37% ♣	38%	4%	21%

Table 5.13: Percentages of user utterances with each time format. The highest proportion for each system condition is highlighted in **bold**. ♠ indicates a statistically significant difference from the **highest value** in the column (p<.05 with Bonferroni adjustment). ♣ indicates a statistically significant difference from the **highest value** in the column (p<.01 with Bonferroni adjustment)

unprimed U_TIME_APM forms for the different system conditions. The proportion of U_TIME_APM forms in *First primed* utterances is significantly higher in the SYS_TIME_APM condition than in the SYS_TIME_ONLY condition (p<.01), although not significantly different than in the SYS_TIME_POD condition. The proportion of U_APM forms in the *All primed* utterances is significantly higher in the SYS_TIME_APM condition than in both the SYS_TIME_ONLY and the SYS_TIME_POD conditions (p<.01). **I conclude that there is user adaptation to the TIME_APM form.**

U_TIME_ONLY

There are no statistically significant differences in the proportions of unprimed U_TIME_ONLY forms for the different system conditions. The proportions of U_TIME_ONLY forms in the *First primed* utterances in the SYS_TIME_ONLY condition is significantly higher than that in the SYS_APM condition (p<.01), but not significantly higher than that in the SYS_POD condition. The same is true of U_TIME_ONLY forms in the *All primed* utterances. **I conclude that there is user adaptation to the TIME_ONLY form.**

U_TIME_POD

I did not find statistically significant differences in U_POD forms for the different system conditions in either the *unprimed*, *First primed* or *All primed* data. The proportions of TIME_POD in user utterances after confirmation is as low as it is before confirmation in the SYS_TIME_POD condition. I note that this is the *long unambiguous* form; users may have felt that it would not be recognized or that it was inefficient to produce it.

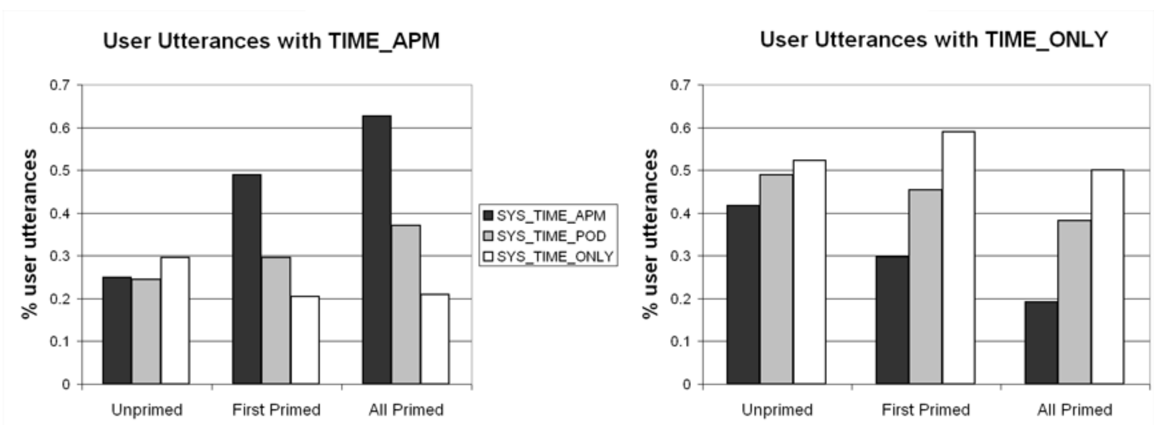

Figure 5.1: Proportions of user utterances with the TIME_APM and the TIME_ONLY for each system condition

Figure 5.1 graphically illustrates the proportions of user utterances with the two time forms (TIME_APM and TIME_ONLY) that exhibit user adaptation effect. After priming users are more likely to use the same condition as the system: the SYS_TIME_APM bar is the highest for TIME_APM utterances (left graph, black bar) and the SYS_TIME_ONLY bar is the highest for the TIME_ONLY utterances (right graph, black bar). I also observe that TIME_APM form decreases after priming in SYS_TIME_ONLY condition (left graph, white bar) while TIME_ONLY form decreases in SYS_TIME_APM condition (right graph, white bar). The proportion of utterances in the SYS_TIME_POD condition behaves similarly to the SYS_TIME_APM condition: it grows after priming for TIME_APM (left graph, gray bar) and decays after priming for TIME_ONLY (right graph, gray bar), although at a lower rate.

5.3.4 Results: Comparing User Adaptation with Previous Work

System condition	keep	switch to adapt	switch to different
System adaptive SYS_APM	APM→APM	-	APM→T/O, APM→POD, APM→CLOCK
System adaptive SYS_TIME_ONLY	T/O→T/O	-	T/O→APM, T/O→POD, T/O→CLOCK
System non-adaptive SYS_TIME_APM	T/O→T/O, POD→POD, CLOCK→CLOCK	T/O→APM, POD→APM, CLOCK→APM	T/O→CLOCK, T/O→POD, POD→CLOCK, CLOCK→POD,
System non-adaptive SYS_TIME_ONLY	APM→APM, POD→POD, CLOCK→CLOCK	APM→T/O, POD→T/O, CLOCK→T/O	APM→CLOCK, APM→POD, POD→CLOCK, CLOCK→POD,

Table 5.14: User action in adaptive and nonadaptive system conditions

In this section I compare my results to the previous work by Brennan (1996). The author analyzed lexical convergence of a user with a Wizard-of-Oz dialog system. The experiment measured convergence of a user to a system's embedded (implicit) and exposed (explicit) corrections of a term, such as *school/college*. Brennan (1996) reports proportion of cases where the user *switches* to use system's term.

In *Let's Go!* system the user can 1) keep the same time form, 2) switch to adapt to a system's form, or 3) switch to a different form. Table 5.14 illustrates the possible user's actions in the *Let's Go!* system. Here, I look at the proportions of user utterances in the *switch to adapt* column of the table. I define *convergence to TIME_APM* to be the proportion of utterances in the SYS_TIME_APM condition that *switch* from another form to TIME_APM. I define *convergence to TIME_ONLY* to be the proportion of utterances in the SYS_TIME_ONLY condition that *switch* from another form to TIME_ONLY.

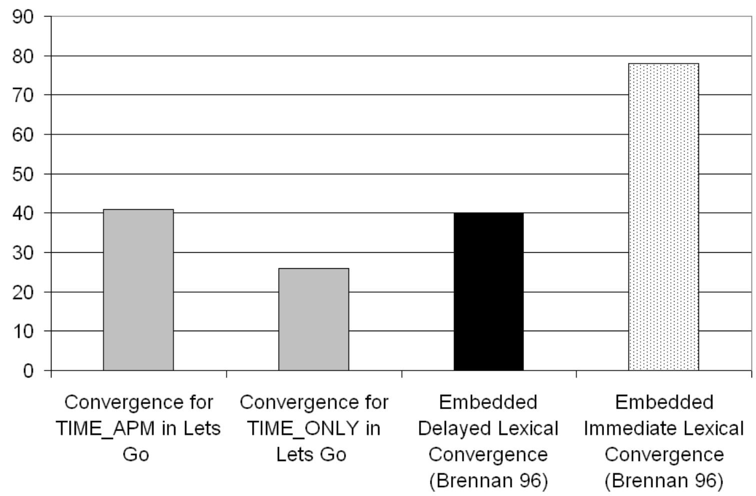

Figure 5.2: Comparing lexical convergence in the *Let's Go!* system and in Brennan (1996)'s experiment

Figure 5.2 shows the proportions of *convergence to TIME_APM* and *convergence to TIME_ONLY* and the proportions of the the embedded delayed and embedded immediate lexical convergence in Brennan's experiments.[11]

In the *Let's Go!* system *convergence to TIME_APM* is higher than *convergence to TIME_ONLY*. This may be due to the difference in the two prompts. TIME_APM is the prompt that primes with presence of AM/PM, while TIME_ONLY primes with absence of any suffix. To *converge to TIME_ONLY* a user would have had to have

[11] Since my experiment is more similar to embedded (or implicit) correction in Brennan's experiment, I only show the results for convergence with embedded correction.

another suffix originally, such as am/pm, o'clock, etc. These results suggest that the priming with a suffix (am/pm) has a stronger effect than priming with no suffix. Perhaps, users are more inclined to switch from using one suffix to another than from using a suffix to no suffix. This difference may also be caused by the ambiguity of the TIME_ONLY form. The priming effect of the TIME_ONLY form conflicts with the ambiguity of the format. In some cases users may want to emphasize the part of the day that they are interested in.

Brennan reports higher convergence in immediate utterances than in delayed utterances. Convergence in the time form in *Let's Go!* system is lower than lexical convergence in Brennan's experiment. Although I measured user's switch of time form immediately of priming, *convergence to TIME_APM* in the *Let's Go!* is comparable to the delayed convergence and is lower than immediate convergence in Brennan's experiment. In Brennan's experiment, the system's embedded correction happens in the system's answer:

User: what **college** does Aida attend?

System: the **school** Aida attends is Williams

while in *Let's Go!* system, the embedded correction happens in a question:

User: seven **o'clock**

System: Did you say one **a. m.**?

Brennan's results show that the users are more likely to adapt to exposed correction than to embedded correction. I hypothesize that the type of utterance containing the embedded correction may affect convergence. When primed by a question that exposes an error in the system's understanding (as in my *Let's Go!* experiment), the user's attention may be shifted to the semantic information that needs to be specified. In this case the user may be less likely to notice the different form used by the system and consecutively less likely to converge. I also hypothesize that the nature of the task may affect convergence. The *Let's Go!* experiment was conducted with real users looking for bus information over a noisy phone channel, while Brennan's experiment was conducted with paid subjects in a laboratory.

5.3.5 Results: the Effect of System Adaptation on the User

condition	keep the same form	switch to adapt to the system	switch to different than the system	total
adaptive system cond.	81.8% (27)	-	18.2% (6)	33
non-adaptive system cond.	36.7% (18)	28.6% (14)	34.7% (17)	49

Table 5.15: Proportions of user actions in *First primed* confirmation utterances (keeping or changing the form of time)

In this section I use the dialogs in the SYS_TIME_APM and SYS_TIME_ONLY conditions for evaluating the effect of the system appearing to *adapt* to the user. *Adaptive cases* include the dialogs when 1) the user in an *unprimed* utterance says TIME_APM in the SYS_TIME_APM condition; and 2) the user in an *unprimed* utterance says TIME_ONLY in the SYS_TIME_ONLY condition (see Table 5.14). Although in this experiment the system did not explicitly *adapt* to the user, the system's behavior (choice of time form) is co-incidentally adaptive in these two scenarios. The TIME_POD form is extremely rare in user utterances. The experiment did not produce *adaptive cases* in the SYS_TIME_POD condition. Hence, I excluded the SYS_TIME_POD condition from this analysis.

The dataset contains 33 adaptive and 48 non-adaptive dialogs for the two system conditions[12]. I examine the *First primed* confirmation user utterance. I differentiate between three possible user actions: 1) keep (the time form is unchanged), 2) switch

[12] I excluded those dialogs where users chose to start a new query after time specification.

to same (the user switches to the system's form), and 3) switch to different (the user switches to a different form from the system's form).

Table 5.15 shows the proportions and the number of cases when the user keeps or switches the *time form*. To test statistical significance of the results I perform inference on proportions test. The results indicate that in the *adaptive* condition users are twice as likely to keep the *time form* than in the *non-adaptive* condition (81.8% vs. 37.5%). This difference is statistically significant ($p<.001$).

In the *non-adaptive* system condition users who switch the *time form* are slightly more likely to switch to a different *time form* (35.4%) than to the system's form (29.1%). The results suggest that when the system does not adapt to the user, user's choice is unpredictable. However, if the system adapts to the user, the user is likely to keep the same form. This means that if the system can adapt to the user when the user chooses a form that is more likely to be recognized correctly, that provides positive reinforcement, making the user more likely to use that felicitous form in the future. Furthermore, if the system does adapt to the user then it may be possible with high accuracy to predict the users form for subsequent utterances, and to use this information to improve ASR accuracy for subsequent utterances (Stoyanchev and Stent, 2009a).

One might argue that users' lack of adaptation in the *non-adaptive* system condition is caused by the semantic difference between the time forms used for priming. TIME_ONLY is an ambiguous form. Consider a case where the user indicates a departure time *four o'clock* in the SYS_TIME_ONLY condition:

S: What time would you like to leave?

U: four o'clock

S: leaving at seven

U: No, at **four p. m.**!

The system's confirmation uses an ambiguous TIME_ONLY form. Why is the user not likely to switch to the TIME_ONLY form? Possibly, the user may be inclined to clarify the part of day using unambiguous form in the *First primed* confirmation utterance. However, users who used TIME_ONLY in the *unprimed* utterance, tend to keep TIME_ONLY in the SYS_TIME_ONLY condition. Hence, this argument does not hold.

Consider an example in the SYS_TIME_APM system condition:

S: What time would you like to leave?

U: four

S: leaving at seven a. m.

U: No, at **four**!

Why is the user not more likely to switch to the TIME_APM form? According to Gricean maxim of quantity (Grice, 1981), a speaker's contribution is "as informative as is required for the current purposes of the exchange". According to the this principle, speakers specify the minimum amount of the information required for the message to be understood. In the case of a correction, the minimal information is only the part needing correction (i.e. *seven*). However, in the *adaptive* SYS_TIME_APM condition, users are more likely to keep the TIME_APM form when primed with TIME_APM. Hence, this argument does not hold.

The result of the *adaptation to time form* experiment contradicts most of the previous experimental results on adaptation that suggest users' adaptation to the system. What is the difference in this case? This experiment is different from the past experiments in two ways. First, I am looking at adaptation to the form of a *concept*, while most other experiments evaluated adaptation to verbs or nouns. Second, the priming happens after a user has had a chance to say one of the *time forms*.

The result suggests that the *adaptive system condition* where the user's choice of the *concept form* is reinforced by system's prompt, increases the user's probability of repeating this form. In this experiment, I expected to find that system priming causes users to change their *time form* to the same form as the system. Instead, I found an effect of *system adaptation* on the probability of *user's change*: system adaptation affects the likelihood of the user changing the *concept form*.

5.4 Discussion

In this chapter, I showed that in deployed dialog systems with real users, as in laboratory experiments, users adapt to the lexical and syntactic choices of the system. I analyzed users' adaptation to the presence of verbs and prepositions, to the verb form, and to the form of a task-related concept. I showed that users do adapt to the system's lexical and syntactic choices, as well as concept forms.

These results indicate that the system prompts have **directive power**, or the ability to guide users into using particular words and syntax. Formulation of system prompts in a flexible input dialog systems can be used to guide users into producing utterances conducive to task success. My results show that variations in system prompts can have an impact on recognition of task-related concepts. The system's the ability to guide a user may depend on multiple factors, such as the amount and type of information contained in the prompt, dialog history, or the user's focus of attention. In future work, I would like to evaluate which factors make it more likely for the users to adapt to the system.

I showed that users are more likely to adapt to the system's choice when the system appears to adapt to them. The finding of the effect of system adaptation to the user has a potential implication for the design of dialog systems: systems should adapt to the user's choices of concept forms. By adapting to the user, the system guides the user into adapting to self and to the system, leading to a more predictable user behavior. The predictable user behavior enables the system to limit its grammar and language model in the ASR and NLU components and potentially improve speech recognition and concept identification in a dialog system. In my experiments I found that users adapt to the form of time concept in the *Let's Go!* bus information system. I hypothesize that presence of adaptation to the form of time concept in the *Let's Go!* suggests that this **directive** effect is likely be present for other types of concepts and needs to be investigated further. In the future work I would like to confirm the effect of system adaptation to other system concepts, such as a *place* concept (in an application where place has multiple realizations). Dialog applications with a more diverse domain than bus information, such as tutoring or technical assistance, may be interesting for studying effect of adaptation as they have more topic-specific (math, physics, etc.) concepts with multiple realizations.

Chapter 6

Responsive Adaptation in Dialog

6.1 Motivation and Research Goal

Responsive adaptation involves a change in a system's behavior in response to a user or a dialog situation. The change can be manifested in any of the components of a dialog system: Natural Language Understanding, Natural Language Generation, Dialog Management, or Speech Recognition, as illustrated in Section 2.3. In this work I address responsive adaptation in the speech recognition module of a dialog system.

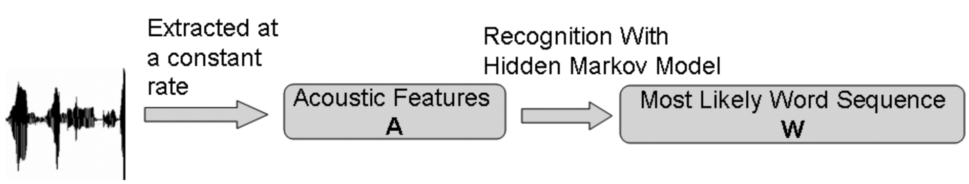

Figure 6.1: Automatic speech recognition

Speech recognition is a statistical process. Acoustic frequency features A are extracted at a constant rate from an utterance (see Figure 6.1). Recognition is achieved by maximizing the probability of the word sequence, W, given the acoustic features, *A*. Speech recognizers use two kinds of models: an acoustic model and a language model. An acoustic model is generated from spoken data with aligned transcription. Acoustic model encodes the probabilities of mapping acoustic frequency features to lexical units. A language model encodes probabilities of n-grams (strings of length n) occurring in an utterance. Modern recognizers commonly use 3-gram models that record probabilities of words (1-grams), bi-grams, and tri-grams. A language model can be statistical (generated from text) or grammar-based (generated from a manually constructed context free grammar). This means that the similarity between a recognized user utterance and the dataset used for generating a language model affects the performance of recognition. For example, an utterance containing words frequent in the language model is more likely to be recognized correctly than an utterance containing infrequent words.

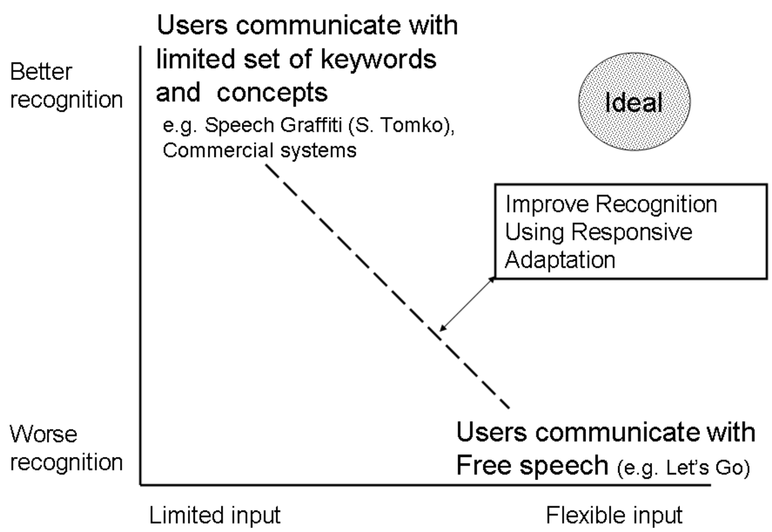

Figure 6.2: Dialog systems recognition and interaction

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). Word error rates in research dialog systems are known to be even lower. Noisy conditions, speaker's accent, or speaking out of vocabulary increase recognition errors.

The performance of speech recognition is often dependent on the type of input the system is designed to recognize (see Figure 6.2). On the one hand, *limited input* dialog systems require the user to respond to each system prompt using only the concepts and words currently requested by the system. On the other hand, *flexible input* dialog systems allow the user to respond to system prompts with phrases and sentences and specify information other than that currently requested. *Flexible input* systems may also allow the user to take task initiative.

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* dialog system achieve better speech recognition than users communicating with *flexible input* dialog systems (Tomko and Rosenfeld, 2006). Sheeder and Balogh showed that in *flexible input* dialog systems prompts can be formulated to maximize ASR accuracy and reduce the number of ASR timeouts (Sheeder and Balogh, 2003).

I hypothesize that information about the content of a user utterance may help improve speech recognition for the utterance. I automatically predict the content of user utterances using features from the dialog content and from the utterance. Then I adapt the ASR's language model to the predicted content of the user's utterance.

It is now common practice to adapt the recognizer to the type, context or style of input speech (Bellegarda, 2004). Language model (LM) 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., 2008a), broadcast news recognition (Gildea and Hofmann, 1999), and spoken dialog systems (Tur et al., 2005). LMs in dialog systems can be adapted to the dialog state (Riccardi and Gorin, 2000; Esteve et al., 2001), the topic (Iyer and Ostendorf, 1999; Gildea and Hofmann, 1999), or the speaker (Tur, 2007). In this work I use the concept type(s) in the user's utterance to adapt the recognizer's LM.

6.2 Experimental Approach

6.2.1 Concepts and Confirmations

In this experiment I address the problem of speech recognition of concepts (system-specific information provided by a user) that are specified by a user after system confirmations (yes/no questions confirming system's understanding of a concept). System's ability to recognize a concept is essential for successful conversation. Failure to recognize a concept may lead to cascading errors and complicate dialog. System may improve its concept recognition by adapting its language model to dialog context. In request prompts, when a system requests specific information, context is the type of information requested. However, after confirmation promts, users often choose to switch context making it more difficult for the system to adapt to context. My work addresses this issue. I show that it is possible to automatically predict dialog context after a system's confirmation and improve speech recognition or user's concepts by adapting language model to the automatically predicted context.

In this experiment I use annotated dialog transcripts and speech from the *Let's Go!* system described in Section 4.2. *Lets Go!* is a telephone-based spoken dialog system that provides information about bus routes in Pittsburgh (Raux et al., 2005). The data used in this experiment comes from the first two months of *Let's Go!* system operation in 2005 (2411 dialogs), and one month in 2006 (1430 dialogs). Researchers at Carnegie Melon transcribed and hand-annotated this data for concept types. In the annotated transcripts, the following concept types are labeled: `neighborhood`, `place`, `time`, `hour`, `minute`, `time-of-day`, and `bus`. For the experiments I collapsed these concepts into three concept types: *time*, *place* and *bus* (see Table 6.1)

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 6.1: Examples of user utterances with a concept in *Let's Go!* system. Concept annotations: **:p** indicates place, **:t** indicates time, and **:b** indicates bus.

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

Table 6.2: Example answers to system confirmations

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, Braddock avenue") or a *topic change* (e.g. "no, leave at 7" or "yes, and go to Oakland"). (See Table 6.2 for more examples of users' responses to confirmation questions in the *Let's Go!* corpus). The user's response type has implications for further system 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. In *Let's Go!* users specify a concept in 18% of post-confirmation utterances. As Figure 6.3 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. My goal is to improve recognition of the user's post-confirmation utterances that contain a concept.

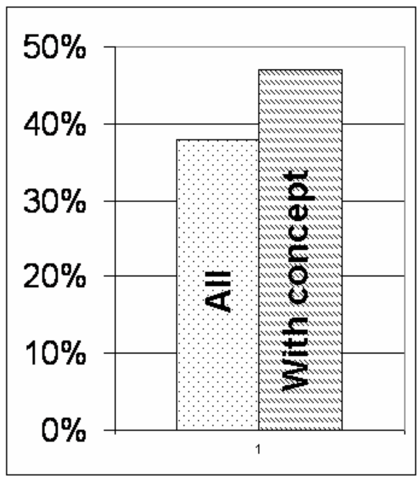

Figure 6.3: Word error rate on post-confirmation user utterances

6.2.2 Dialog States and Language Models

In order to provide the user with route information, *Let's Go!* elicits a departure location, a destination, a departure time, and optionally a bus route number. *Let's Go!* has four dialog states corresponding to the information it elicits: *first-query*, *place*, *time*, and *confirm*. Figure 6.4 illustrates the dialog states used in the original version of *Let's Go!*. Each concept value provided by the user is explicitly confirmed by the system.

In each of the system states, a state-specific language model is used for recognizing a user's answer. The state-specific language models are trained on user utterances

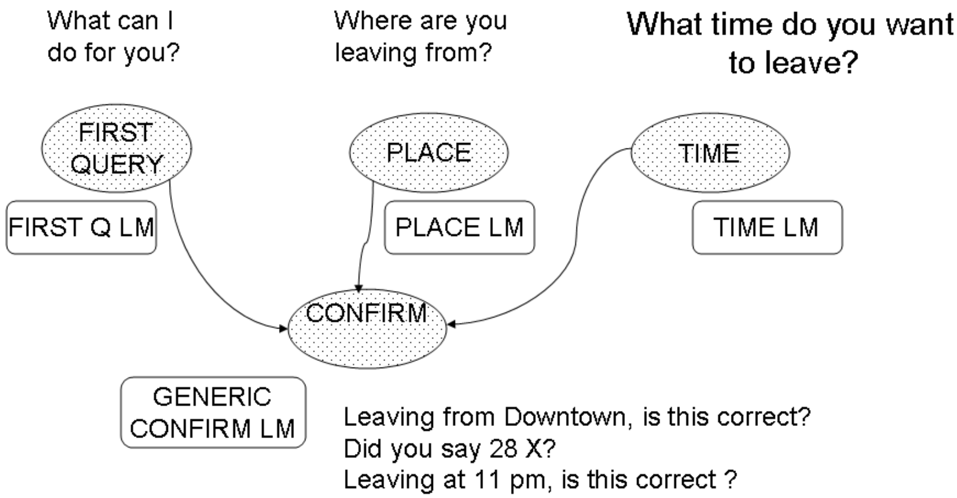

Figure 6.4: Dialog states and language models used in the original version of *Let's Go!*

from the corresponding system states in the 2005 dataset. State-specific language models reflect the distribution of user's answers in each state. For example, the `Place LM` is built on user's answers to the *Where are you leaving from?* prompt. `Place LM` has a higher chance of recognizing correctly typical answers with vocabulary such as *leaving, from, going, to* and a name of a place.[1] It is less likely to correctly recognize untypical answers (e.g. *at four*).

The *generic Confirm* LM is trained on all user post-confirmation utterances from the 2005 dataset. Typical answers to confirmation prompts (e.g. *Leaving from X?, Traveling at Y?, You want the bus Z?*) do not contain a concept. Only 15.6% of post-confirmation utterances in the 2005 dataset contain a *place* concept, 3.2% contain a *time* concept, and 6.4% contain a *bus* concept (see *Concept Type Features* in Table 6.3). Hence, utterances with a concept are not well represented by the *Generic Confirm LM* and recognition is likely to fail on utterances containing a concept. Even though such

[1] Language models used by *Let's Go!* are hierarchical, the concept names are stored in the dictionary. If training data contains an utterance with place concept (labeled with :p), the model should be capable of recognizing all places in the database used in a similar context.

utterances are quite rare, they are disproportionately important. This work addresses the relatively small problem of improving recognition of post-confirmation utterances with a concept (18% of all post-confirmation utterances). However, misrecognition of a concept is a critical problem in a dialog system that can lead to cascading errors. My previous analysis of 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 and Stent, 2009a). Correct determination of the concept type of post-confirmation utterances can lead to improved speech recognition, fewer and shorter sequences of speech recognition errors, and improved dialog system performance.

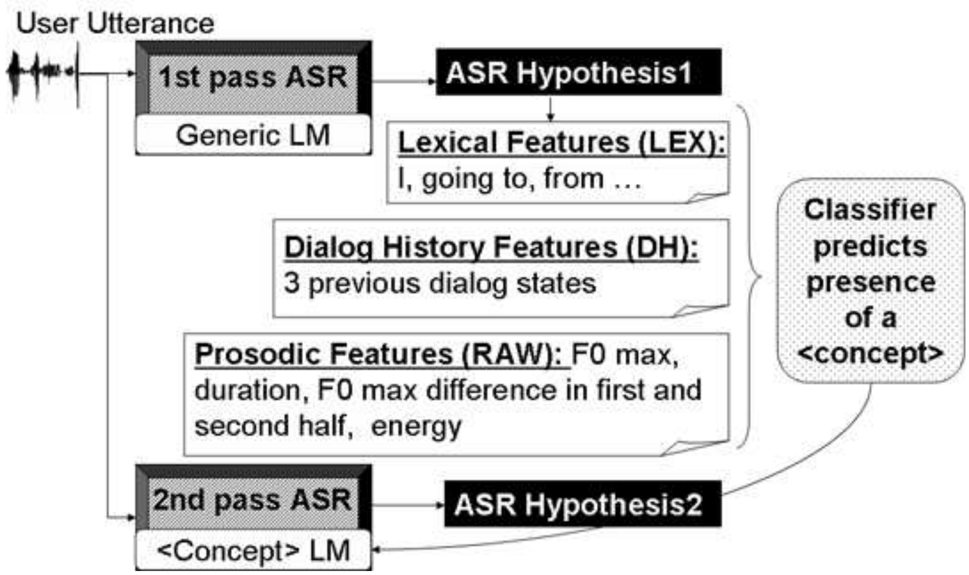

Figure 6.5: Two-pass Automatic Speech Recognition approach

I adopt a two-pass recognition architecture previously introduced by Young (1994). The process is shown in Figure 6.5. In the first pass, the input utterance is processed using the `generic confirm LM`. 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"). I 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. I 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.

6.2.3 Post-Confirmation User Utterances in *Let's Go*

Table 6.3 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 are significantly different. Most confirmation prompts in both data sets are for a *place* (61% and 59.2% respectively). However, in the 2005 dataset the *bus* and *time* concepts occurred with almost the same frequency in confirmation prompts (19.4% and 19.6%), while in the 2006 dataset, *bus* concepts occurred in only 17.6% of confirmation prompts and *time* concepts in 22.3% of confirmation prompts. Perhaps some users figured out that bus is actually not a required piece of information; start and end locations are sufficient for the system to figure out the bus route. There are also differences in user responses to confirmation prompts. The proportion of responses containing "yes", "no", and/or a concept all dropped from the 2005 dataset to the 2006 dataset. This may be caused by users in the 2006 dataset using more variation when responding to confirmation prompts.

I also observe some differences in variance of duration of users' utterances. This may be due to improvement in detecting when a user stops speaking. The 2006 dataset also shows higher RMS mean that may be due to change in the hardware settings in the two years of operation.

Event	2005		2006	
	num	%	num	%
Total dialogs	2411		1430	
Total confirm utts	9098	100	9028	100
Confirms utts with a concept	2194	24	1635	18.1
Dialog 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 Features				
feature	mean	stdev	mean	stdev
Duration (seconds)	1.341	1.097	1.365	1.242
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 6.3: Statistics on post-confirmation utterances

Because of these differences between two datasets, I used cross-validation on the 2006 data for the concept type classification experiments. In my experiments, I used the 2006 data to train concept type classifiers and for testing. I used the 2005 data to build LMs for the speech recognition experiment.[2]

[2] I chose to use the 2005 dataset for building language models to use more data for training. The difference between the datasets may lower the speech recognition performance across all of the experiments.

6.3 Predicting Concept Type

In this section I describe my experiments on concept type prediction.

6.3.1 No-Concept Baseline Prediction

Since majority of the post-confirmation utterances do not contain a concept, the first baseline method predicts "no concept" for all utterances. 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.

6.3.2 Confirm-Type Baseline Prediction

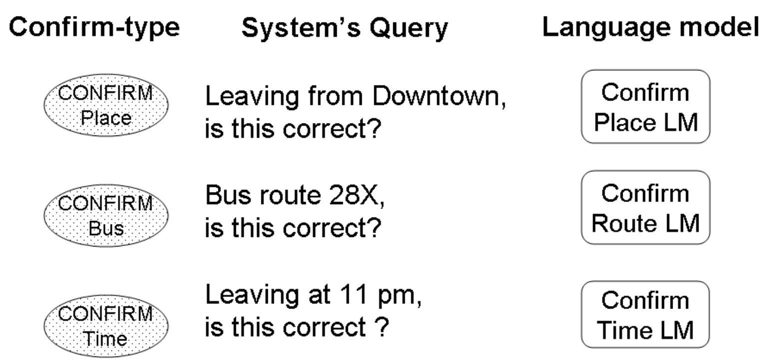

Figure 6.6: A confirm-type baseline approach to language modeling

A simple approach to predicting concept types in user utterances is to use the concept type being confirmed by the system (Figure 6.6). If the system requests confirmation of a *place*, this method predicts that the user's post-confirmation utterance will contain a *place* concept. If the system requests confirmation of a `bus`, it predicts a *bus* concept. If the system requests confirmation of a `time`, it predicts a *time* concept.

There are two problems with this approach. First, the majority of utterances (82%

	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 6.4: Confirmation state vs. user concept type

in 2006 dataset) do not contain any concept. Second, users may attempt topic changes in post-confirmation utterances, or use a different concept than the one confirmed. Table 6.4 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. Post-confirmation corrections are more likely to be topic changes in the 2006 dataset than in the 2005 dataset. In the Section 6.4.2 I compare speech recognition results using prediction of the **confirm-type** baseline method with the machine learning method.

6.3.3 Machine Learning Method

I use decision trees to classify each post-confirmation user utterance by the concept type(s) it contains (*place, time, bus* or *none*). I experimented with using lexical, prosodic, and dialog history features in the machine learning algorithm. All of these features are available at run-time and can be used in a live system. The features are outlined in Table 6.5 and described below.

System confirm-type feature (DIA)

The system confirm-type feature corresponds to the *confirm-type* baseline prediction. It indicates the concept type requested in the confirmation prompt and takes the

Feature type	Feature source	Feature description
System confirm-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(*first query, place, time, confirm_place, confirm_time, 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 6.5: Features for concept type classifiers

value of *place*, *bus*, or *time*.

Acoustic Features (RAW)

The acoustic features are extracted from the raw audio of the user utterances. I use utterance maximum pitch (F0 max), energy (RMS), duration, and the difference between F0 max in the first and second halves of the utterance. The F0 difference feature is intended to capture the raising and falling intonation in a user's utterance. I used Pratt (Boersma and Weenink) scripts to automatically extract these features from the audio. These features were inspired by the work of Litman et al. (2006) on detecting speech recognition errors. I anticipated that these features would help distinguish corrections and rejections from confirmations.

#	spkr	Utterance	DIA	State History	Concept History
1	S	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 **downtown**	conf bus	SH1=first_query SH2=∅ SH3=∅	CH1=bus CH2=∅ CH3=∅
5	S (conf)	Leaving from the Airport. Is this correct?			
6	U (post conf)	Yes.	conf place	SH1=confirm_bus SH2=first_query SH3=∅	CH1=place CH2=bus CH3=∅
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=∅ CH2=place CH3=bus

Table 6.6: Dialog state and history features example

Dialog History Features (DH)

I use from one to three utterances of dialog history (DH1, DH3). These features capture information about the dialog state history (SH) and concept history (CH). In DH1, the dialog state and concept of the previous utterance are recorded. In DH3, the dialog states and concepts of the three previous utterances are recorded. The dialog state values can be *first query, place, time, confirm_place, confirm_time,* or *confirm_bus.* This feature is extracted from the system log of the dialog. The concept history values are extracted from the annotated, transcribed speech and could be place, time, bus, or none.[3] Figure 6.6 shows the values of dialog state (DIA), state history, and concept history features on an sample from a *Let's Go!* dialog. User's utterance #4 is an answer 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.

Lexical Features (LEX)

LEX features include non-concept words and bigrams from the user's current utterance, such as *go, leave, to, from, etc.* I hypothesize that these features are highly indicative both of concept presence and absence, 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. I explored two methods for identifying the most salient lexical features: manual and mutual information extraction. Both of these methods selected a set of the most salient features that were then used for concept classification.

Manual approach:

I 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

[3]Concept history can have multiple values. It is represented as a binary feature for each of the possible concepts (*bus*, *place*, or *time*).

`I` (indicates complete sentence). These features were selected based on a heuristic estimate of their importance and their high relative frequency in the corpus.

Mutual information approach:

I selected lexical features according to the *mutual information* between potential feature and concept types (Manning et al., 2008). I extracted lexical features (unigrams and bigrams) from the transcribed user utterances. I removed all words that realize concepts (e.g. "61C", "Squirrel Hill"), as these are likely to be misrecognized in the first pass recognition of a post-confirmation utterance. I computed the mutual information between each potential lexical feature and concept type and selected features with the highest mutual information score.

I 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 6.7 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
route					
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 6.7: Mutual information for selected features

I try two methods of selecting features with the highest MI. In the first method I select for each concept type the 50 features with the highest mutual information. In the second method I select for each concept type the 30 features with the highest mutual information that occurred at least 20 times in the training data[4].

Concept Type Match Features (CTM)

The CTM feature indicates whether a user's utterance matches a concept. I tokenized all concepts: names of bus stops, places, buses, and time. Each automatically recognized user utterance was matched to the bag of words for each of the concepts. I used three binary features CTM_*place*, CTM_*bus*, and CTM_*time*. For example, CTM_*place* feature is set to **true** when a recognized utterance matches a part of one of the *place* concepts, such as *street* or *avenue*.

For transcribed speech there is a one-to-one correspondence between presence of

[4]I aimed to select an 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.

the concept and the CTM feature. So this feature alone has 100% concept prediction accuracy. Hence I only evaluate this feature for recognized speech. I hypothesized that the CTM feature will improve cases where a part of (but not the whole) concept instance is recognized in first-pass recognition. 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* 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.

6.3.4 Experimental Results

In this section I examine the impact of the features presented in Table 6.5 on concept type classification performance. I report overall classification performance separately for feature combinations with lexical features from transcribed speech (Table 6.10) and from recognized speech (Table 6.12). The results on transcribed speech gives us an idea of the best possible performance on concept type classification[5]. The results on recognized speech provide a realistic estimate for the performance in a live dialog system.

I performed a series of 10-fold cross-validation experiments to examine the impact of different feature combinations on concept type classification. I trained three binary classifiers for each experiment, one for each concept type, i.e. I separately classified each post-confirmation utterance as *place +* or *place -*, *time +* or *time -*, and *bus +* or *bus -*. I used Weka's implementation of the J48 decision tree classifier (Witten and Eibe, 2005)[6]. The overall performance is computed over all three concepts.

[5] I exclude concept words (e.g. Downtown) from LEX features.

[6] J48 gave the highest classification accuracy compared to other machine learning algorithms I

Performance for predicting each concept is reported in Table 6.11 for transcribed speech and Table 6.13 for recognized speech.

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

Table 6.8: Measures of concept prediction. tp=True Positives, tn=True Negatives, fp=False Positives, fn=False Negatives

For each experiment, I 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*). I do not report precision or recall for determining *absence* of each concept type. In the data set 82.2% of the utterances do not contain any concepts (see Table 6.3). Consequently, precision and recall for determining absence of each concept type are above .9 in each of the experiments. I also report overall *pre+*, *rec+*, f-measure (*f+*), and classification accuracy across the three concept types. Finally, I report the percentage of *switch+* errors and *switch* errors. *Switch+* errors are the proportion of utterances with a concept classified as containing a different concept. 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

tried on this data.

concept different than the concept in the utterance and will be likely to have a higher word error rate. Utterances with a concept misclassified as *none* will be decoded with the same *generic confirm* language model in the second pass of the recognition. The word error rate while recognizing these utterances in the second pass will be the same as in the first pass. The *Switch* error is the proportion of all utterances misclassified with one of the concepts. *Switch* errors include utterances with no concept classified as *place*, *bus* or *time*. Table 6.8 outlines each of my performance measures and describe how they are computed.

I compare results using a paired t-test with Bonferronni correction. 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+*) 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 the ASR. 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 may cause speech recognition performance to suffer. This means that I want to maximize (*rec+*) and minimize *switch+* errors.

Baselines

The **No-Concept** baseline achieves overall classification accuracy of 82% but *rec+* of 0. At the other extreme, the **Confirm-type** baseline achieves *rec+* of .79, but overall classification accuracy of only 14%. I always use the current confirmation prompt type (DIA) feature.

Features	Classification accuracy	
	rec+	acc
$\text{LEX}_{manual5}$	0.55	0.89
$\text{LEX}_{topMI50}$	0.52	0.88
LEX_{freq30}	**0.56**	**0.89**
$\text{RAW+DH+LEX}_{manual5}$	0.57	0.89
RAW+DH+LEX_{50}	0.56	0.89
$\text{RAW+DH+LEX}_{freq30}$	**0.62**	**0.90**

Table 6.9: Comparing selection methods of lexical features. Classification accuracy on lexical features from recognized speech.

Comparing lexical feature selection methods

First, I compare a manually selected lexical features with automatically selected lexical features. I tried two methods for automatic selection of lexical feature sets: (a) LEX_{50}, the 50 features with the highest mutual information; and (b) LEX_{freq30}, the 30 features with the highest mutual information that occurred at least 20 times in the training data. As Table 6.9 shows, the LEX_{freq30} feature set achieves the highest classification accuracy and rec+. The prosodic (RAW) and dialog history (DH) feature sets lead to additional improvements in performance.

In the experiments described later in this section, all LEX features are selected with the LEX_{freq30} method. Throughout this section, I call the model trained on LEX features the *LEX model*, the model trained on RAW features, the *RAW model*, and so on. The significance tests in this section are done using inference on proportion of correctly classified utterances.

Features from the current utterance (RAW, LEX, LEX_RAW)

I first look at lexical (LEX) and prosodic (RAW) features from the current utterance. A model trained on RAW features achieves *rec+* of 0.34 and overall accuracy of 0.85. This model performs surprisingly well, beating both baselines in overall accuracy

Features	Overall					
	pre+	rec+	f+	acc	switch+	switch
No Concept	0	0	0	**0.82**	0	0
Confirm-type	**0.14**	**0.79**	**0.24**	0.14	0.170	0.723
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**
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 6.10: 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	0	0	.86	0	0	0.81	0	0	.92
Confirm-type	0.87	0.85	0.86	0.64	0.54	0.58	0.71	0.87	0.78
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
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 6.11: Concept type classification results for each concept: transcribed speech (all models include feature DIA).

Features	Overall					
	pre+	rec+	f+	acc	switch+	switch
No Concept	0	0	0	**0.82**	**0**	**0**
Confirm-type	**0.14**	**0.79**	**0.24**	0.14	0.170	0.723
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
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
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 6.12: 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	0	0	.86	0	0	0.81	0	0	.92
Confirm-type	0.87	0.85	0.86	0.64	0.54	0.58	0.71	0.87	0.78
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
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
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 6.13: Concept type classification results for each concept type: recognized speech (all models include feature DIA).

(0.85 vs. 0.82 & 0.14 for the *no-concept & confirm-type* baselines, $p < .001$ for both). However, this model only works for *place* concepts. As shown in Figure 6.13, the *rec+* for 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 with *place* concept contain prosodic information that can be used for classifying presence of a concept. For utterances with *time* and *bus* concepts prosodic features alone are not helpful for determining presence of a concept. One possible reason for this difference in performance may be the lack of training data for the *bus* and *time* concepts. Another reason may be the difference in length of the concept types. Table 6.14 shows average number of non-concept words in an utterance, average number of words in a concept, and average number of characters in a concept[7]. The number of words in utterances and concepts are similar. However, the *time* concept is much shorter in character length[8]. This may explain the low performance of the RAW model on the utterances with the *time* concept, as the prosodic features may not be as reliable on shorter utterances. However, the performance of RAW model on the utterances with the *bus* concept is as low as it is on the utterances with the *time* concept despite the bigger character length of the *bus* concepts. I hypothesize that when users specify a *bus* concept after a confirmation, the values of the prosodic features chosen for this experiment are not different from the values in the utterances without a concept. Hence RAW model is not able to differentiate utterances with the *bus* concept.

concept type	average non-concept# words in utt	average # words in concept	average # char in concept
place	1.29	2.2	12.8
bus	1.63	2.9	10
time	1.73	1.7	6.6

Table 6.14: Length of user utterances with concept

[7]I use the number of characters to approximate the number of syllables

[8]It is not surprising that the time concept is so short. The most common time concept, *now*, is 3 characters long.

LEX model for both transcribed & recognized speech achieve significantly higher *rec+* than the RAW model(0.72 & 0.56 vs. 0.34) and overall accuracy (0.93 & 0.89 vs. 0.85, $p < .001$ for both). As expected, lexical features, even with speech recognition errors, are more useful than prosodic features in isolation. For recognized speech, the LEX model has significantly more *switch+* errors than the RAW model (0.064 vs. 0.099, $p < .001$). This is not surprising since the RAW model has low recall on utterances with concepts (*rec+*), so the majority of errors made by the RAW model are labeling an utterance with a concept as *none*.

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. Prosodic features do not improve the prediction performance. However, lexical features from recognized speech are noisy, so concept type classification for ASR output can be improved by using acoustic/prosodic features.

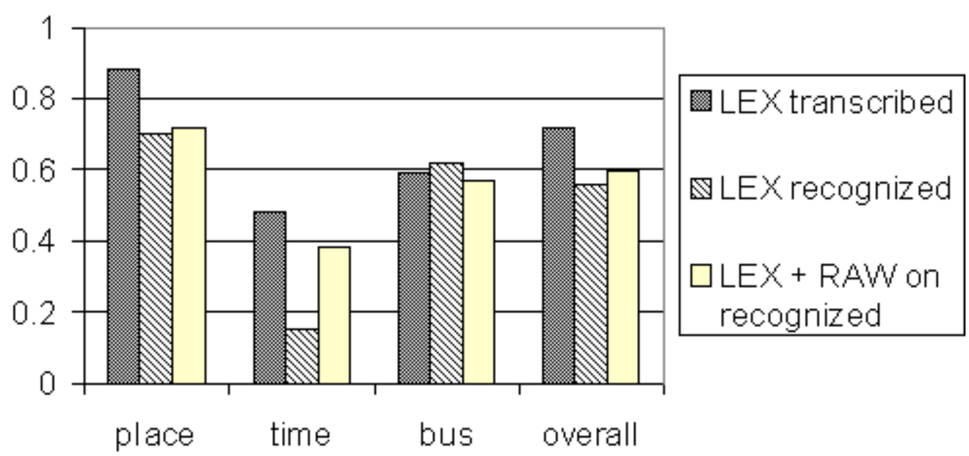

Figure 6.7: Dialog systems' recognition and interaction

Prediction accuracy varies widely across concepts. Figure 6.7 illustrates *rec+*

for *place, time,* and *bus* concepts using LEX from transcribed speech, LEX from recognized speech, and LEX_RAW from recognized speech. Classification of the *place* concept achieves the highest *rec+* out of all of the concepts using each of the feature combinations. This may be partially due to the fact that I have more training data for the *place* concept than for the other concepts, and partially due to more informative lexical features in utterances with the *place* concept. The *Time* concept has the lowest *rec+*, and the biggest drop in performance due to recognition errors (difference between LEX on transcribed and LEX on recognized speech). However, I observe that prosodic features help the *rec+* for *time* concept, improving *rec+* from a low 0.15 to 0.38.

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 ($p < .001$) .

Features from the Dialog History (DH1, DH3)

Next, I add features from the dialog history to my best-performing models so far. For transcribed speech, 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, $p < .001$). DH3_LEX performs significantly worse than DH1_LEX in terms of *rec+* (0.78 vs. 0.81 $p < 0.05$). 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.

Features Specific to Recognized Speech (ASR, CTM)

Finally, I add the ASR and CTM features to models trained on recognized speech.

I 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 corresponds to the finding of Lemon and Konstas (2009) who also find that ASR scores are not helpful for the classification of hypothesis quality.

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 CTM_DH3_LEX_RAW and CTM_ASR_DH3_LEX_RAW.

Summary and Discussion

In this section I 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 ($p < .01$).

6.4 Speech Recognition Experiment

In this section I report the impact of concept type prediction on recognition of post-confirmation utterances in *Let's Go!* system data. I hypothesized that speech recognition performance for utterances containing a concept can be improved with the use of concept-specific LMs. I evaluate speech recognition of post-confirmation user utterances using several types of language models and several approaches to concept prediction. I (1) compare the existing *generic-confirm* LM used in *Let's Go!* with the proposed *concept-specific* adaptation; (2) compare two methods for selecting user utterances for building language models; and (3) evaluate the impact of different concept type classifiers on *concept-specific* LM adaptation.

6.4.1 Method

I 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). I trained trigram LMs using 0.5 ratio discounting with the CMU language modeling toolkit (Xu and Rudnicky, 2000)[9]. I built state- and concept-specific LMs from the *Let's Go!* 2005 data. The LMs encode semantic information (Ward and Issar, 1994b), smoothing probabilities for the concepts not used in the data. I 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.

Method	Models	Data for building a model
Baseline	*generic-confirm*	all post-confirmation utterances
Confirm-type	*confirm-place*	post-confirmation utts after*confirm_place*
	confirm-time	post-confirmation utts after*confirm_time*
	confirm-bus	post-confirmation utts after*confirm_bus*
Concept-based	*concept-place*	post-confirmation utts with *place* concept
	concept-confirm	post-confirmation utts with *time* concept
	concept-confirm	post-confirmation utts after*bus*
	generic-confirm	all post-confirmation utterances

Table 6.15: Methods of building language models

[9]I 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).

Method	Prediction	Decision based on
Baseline	no prediction	
Confirm-type	*confirm-place*	post-confirmation utts after *confirm_place*
	confirm-time	post-confirmation utts after *confirm_time*
	confirm-bus	post-confirmation utts after *confirm_bus*
Concept-based	*concept-place*	Classifier predicts *place* concept
	concept-confirm	Classifier predicts *time* concept
	concept-confirm	Classifier predicts *bus*
	none	Classifier predicts *none* or multiple concepts[10]

Table 6.16: Methods of choosing language models

Language models

I use the language model types outlined in Table 6.15. The *generic-confirm* model is trained on all utterances in the 2005 dataset that were produced in the *confirm* dialog state. This corresponds to the current approach used in *Let's Go!*. The confirm-type models are trained using all utterances from the 2005 dataset that were produced in the *confirm* dialog state following *confirm_place*, *confirm_bus* and *confirm_time* system confirmation prompts respectively. The *concept-based* models are trained on all utterances from 2005 dataset that were produced in the *confirm* dialog state and contain a mention of a *place*, *bus* or *time*.

I use the three methods for choosing language models outlined in Table 6.16. The first, baseline method simply uses one model for recognizing all utterances. For the second method (*confirm-type*) I use the concept-type based confirm-type baseline method to choose one of the three models: *confirm-place, confirm-time* and *confirm-bus*. The third method of choosing a LM (*concept-based* method) uses one of the classifiers described in the Section 6.3. The classifier predicts *place, time, bus*, or *no concept*.

Recognizers

I report results for seven experimental conditions (see Table 6.17). The experimental conditions vary in method of building and choosing LMs. In the experimental conditions 1 - 3, the recognition decoding is done in a single pass. In the **baseline** experimental condition (1), I use the *generic-confirm* LM to recognize all post-confirmation utterances. In the **1-pass confirm** experimental condition (2) I use the confirm-type method for building and choosing language models. I build *confirm-place, confirm-bus* and *confirm-time* LMs to recognize testing utterances produced following a *confirm_place*, *confirm_bus* and *confirm_time* prompt respectively[11]. In the **1-pass concept** experimental condition (3) I 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 experimental conditions 4 - 7 I use the 2-pass recognition model outlined in Figure 6.5. I perform first-pass recognition using the *Generic-Confirm* LM. Then, I classify the output of the first pass using a concept type classifier. Finally, I 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[12]. I used the three classification models with highest overall *rec+*: DH3_LEX_RAW (4), ASR_DH3_LEX_RAW (5), and CTM_ASR_DH3_LEX_RAW (6). To get an idea of "best possible" performance, I also report 2-pass oracle (7) recognition results, assuming an oracle classifier that always outputs the correct concept type for an utterance.

[11] As shown in Table 6.4, most, but not all, utterances in a confirmation state contain the corresponding concept.

[12] I 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.

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

6.4.2 Experimental Results

In Table 6.17 I 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. My goal is to maximize concept recall and minimize WER on concept-containing utterances, without causing overall WER to decline.

As Table 6.17 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 ($p < .01$). *Confirm-type* prediction method has the highest *switch+* (17%) and *switch* (72%) errors (see Table 6.10). With *confirm-type* prediction all utterances without a concept (82%) are decoded with a language model built on utterances with a concept. This explains the increase in overall WER. The *switch+* error indicates that 17% of utterances with concepts were classified with a different concept and decoded with a LM built for different concept. The data used for building LM is non-representative of the data used for decoding. Hence, the WER on these utterances is expected to be higher than the WER with the *Generic-Confirm* model (baseline). This explains the increase in WER on utterances with concept.

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 ($p < .01$).

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

6.5 Discussion

In this chapter, I examined user responses to system confirmation prompts in task-oriented spoken dialog. I 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, I was able to classify task-relevant concepts in the ASR output for post-confirmation utterances with 90% accuracy. I 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 the general language model, I get a fairly high overall WER of 38.49%. In future work, I 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 impact on dialog completion times and task completion rates, as they reduce the number of cascading recognition errors in the dialog (Shin et al., 2002). Furthermore, I 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, I will look at (1) extending the use of the concept-type classifiers to utterances following any system prompt; and (2) the impact of these interventions on overall metrics of dialog success.

In Chapter 5 I described an experiment on directive adaptation where I analyzed user's adaptation to a selected set of choices in system prompts. The directive and responsive adaptation experiments differ in their granularity. In the directive adaptation experiment I analyzed adaptive behavior of human users. Hence, it was useful and feasible to create a fine-grained expereiment by measuring adaptation to selected features. On the other hand, in the responsive adaptation experiment I adapted the recognizer's language model as a whole and analyzed the system's speech recognition improvement. The two experiments differ because of their diverse goals: directive adaptation experiment described in Chapter 5 evaluated user's adaptation and responsive adaptation experiment described in this chapter evaluate the effect of the system's adaptation.

Modern dialog systems adapt language models to static context defined by the system's question. In my approach I evaluated a novel idea of adapting the speech recognizer's language model to dynamic context of a dialog. I have shown that dialog history and prosodic features helped improve automatic prediction of context. The use of prosodic features was motivated by the previous work of Litman et al. (2006) where the authors have shown that prosody can predict users' corrections. During a user's conversation with a dialog system, probability distribution of possible dialog topics mentioned in a user utterance, user's lexical and syntactic choices change dynamically throughout the dialog. I showed that a dialog systems can benefit from dynamically adapting language models and grammars to dialog context and user model on the fly throughout the dialog.

In my experiment I pre-built language model before system execution and predicted which concept is used in a user's utterance. This approach was tractable because the *Let's Go!* system had only three components. I hypothesize that in more complex dialog systems with a larger and more diverse set of concepts, adaptation to the context will also be beneficial for the system. Instead of using a discrete prediction method, a dynamic model adaptation should be done. The ASR language model can be changed dynamically based on dialog context, user's prosody, and system expectations. The work described in this chapter is the first step towards creating flexible and dynamic ASR input components in a spoken dialog systems.

Chapter 7

Responsive Adaptation in Spoken Question Answering

7.1 Motivation and Research Goals

In the previous chapter I looked at responsive adaptation in the speech recognition component of a dialog system. I adapted the ASR's language model to the expected concept in the user's utterance. In this chapter I address speech recognition in spoken question answering, a task closely related to spoken dialog.

Question answering (QA) is the task of automatic retrieval of an answer given a question (e.g. *Who invented silly putty?* or *When was Mozart born?*). Question answering provides a natural language interface for information retrieval. This interface also opens the possibility of access to information retrieval using voice. The user of a spoken-input question answering system may be a reporter who needs to check a fact, a driver on the go, a researcher in the field, or a person with visual disabilities. Spoken question answering can be seen as a more sophisticated version of spoken information access systems such as phone-based directory assistance (Kellner et al., 1998) or weather/restaurant/flight/hotel information systems (Zue et al., 2000).

In this work I address open-domain question answering supporting input questions on a wide range of topics. This task is also addressed in the annual TREC competition (Dang et al., 2006). In the spoken-input question answering task, a question has to be first recognized. This recognition has to be open-domain as input questions may cover a wide range of topics. Word error rates for the state-of-the-art open-domain speech recognition technology are around 25%-30% (Riccardi and Hakkani-Tür, 2003). Goldwater et al. (2008) report that "low-probability words have dramatically higher error rates than high-probability words". This finding indicates that atypical questions containing low-probability words will have lower recognition rates than typical questions with high-probability words.

Question	In what film is Gordon Gekko the main character?
Named Entity	Gordon Gekko
Function Words	in, what, the
Content Words	film, character

Table 7.1: Question components

I address the speech recognition problem for questions containing a **named entity**: a name, a location, or an organization.[1] The words in a question can be classified into one of three categories: named entity, function words, and content words (Table 7.1). Named entities are strongly associated with certain content words. For example for the named entity *Gordon Gekko* related content words are associated with the movie industry, e.g. *film* and *character*. My goal is to improve recognition of these content words using the named entity.

I propose and evaluate a method for improving speech recognition in speech-input question answering system by allowing interaction during the question specification phase. Table 7.2 illustrates non-interactive and interactive approaches to question

[1]Almost all questions in the TREC dataset since 2005 are of this type.

Non-interactive QA	Interactive QA
S: Please say the question.	S: Please say the main topic of your question.
U: In what film is Gordon Gekko the main character?	U: Gordon Gekko
Recognize with open-domain LM	*Recognize with grammar of named entities*
High chance of misrecognizing rare words	S: Please say a question about Gordon Gekko.
	U: In what film is Gordon Gekko the main character?
	or In what film is he the main character?
	Recognize with a LM build from documents matching "Gordon Gekko"

Table 7.2: Interactive and non-interactive question answering approaches

specification. In the non-interactive approach the system recognizes the user's question with an open-domain language model. Rare words have a high chance of being misrecognized. In the interactive approach, the user is first asked to specify the named entity of interest: a person's name, an organization, and so on. A grammar for named entities is created from a database of named entities existing in the target corpus. If a named entity is recognized, a language model specific to the name is used by the speech recognizer. The interaction allows the system to dynamically change language models based on the target named entity, and so to recognize the question's content words better than an open-domain language model. My experimental results show that interactivity feature improves speech recognition performance for spoken-input questions answering system.

Although this approach to question specification may seem awkward, in most question answering evaluations (such as TREC or GALE Distillation) the named entity in consideration is provided in an explicit way. For example in TREC, first the target named entity is given and then several questions are asked about the target. Similarly in the GALE Distillation task, the questions are organized in templates such as *Describe attacks in [LOCATION]* where the variable portion is the named entity. This is in parallel to my design of first getting the name in question.

In the previous chapter I described an approach for improving recognition of spoken responses by adapting language model to a concept. In this chapter I describe an approach for improving recognition of spoken queries by adapting the language model to the named entity in the query.

7.2 Question Answering (QA) System

7.2.1 System Architecture

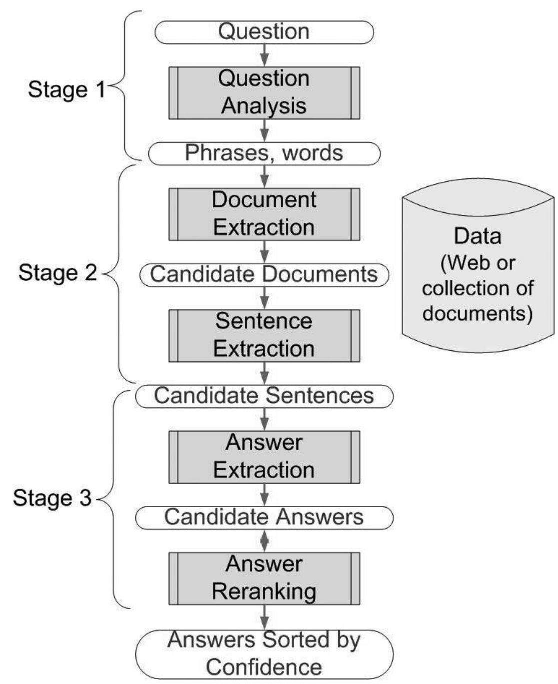

Figure 7.1: Question answering architecture

In this section I describe how QA systems work using the example of the *StoQA* system (Stoyanchev et al., 2008b) developed at Stony Brook. Most question answering systems employ a pipeline architecture with three main stages as illustrated

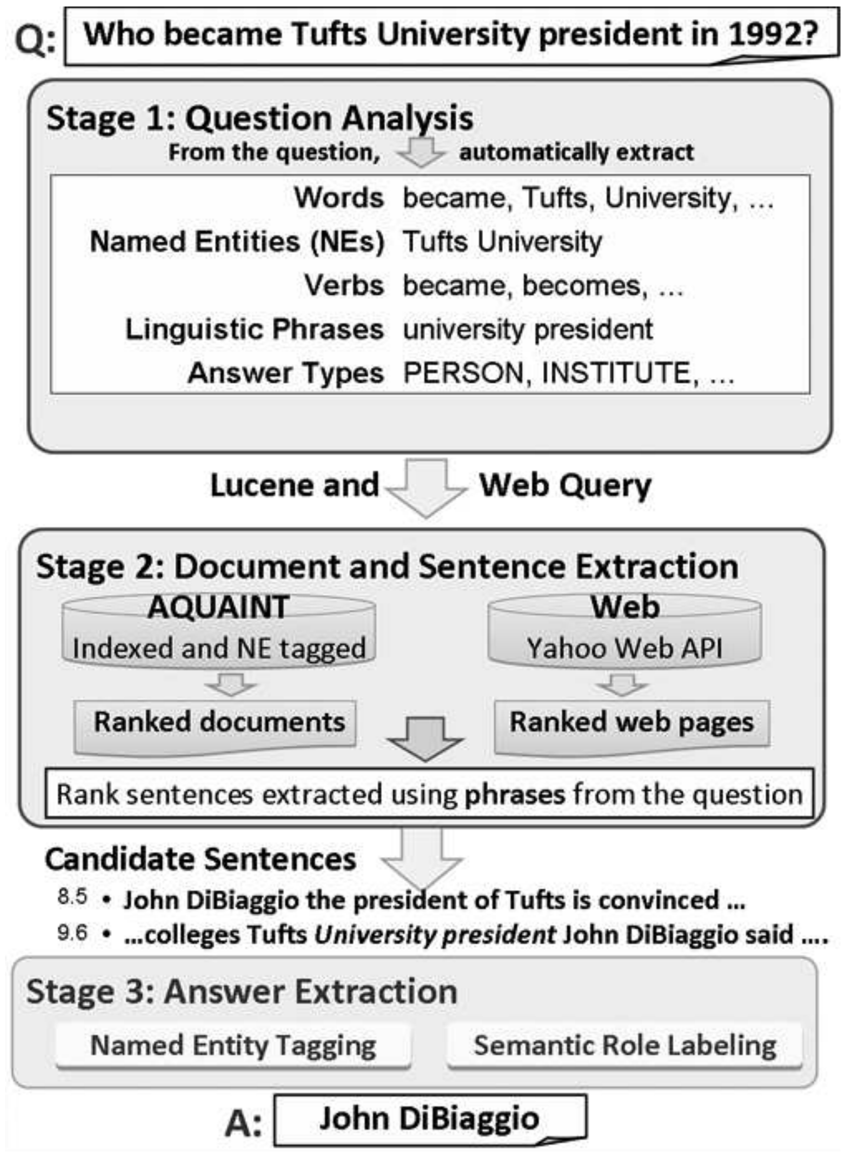

Figure 7.2: Question answering example

in Figure 7.1: question analysis (stage 1), document and sentence extraction (stage 2), and answer extraction (stage 3). Figure 7.2 illustrates an example of automatic answering of the TREC question *Who became Tufts University president in 1992?*.

The question analysis phase involves automatic syntactic and semantic processing of a question and identificatin of question constituents. The output of question analysis is a query tailored to the search tools used in the system. For example, in Figure 7.2 the question analysis phase identifies a named entity *Tufts University*, a verb *became*, a linguistic phrase from the question *"university president"*. The expected entity types for the answer of the illustrated question are *person* or *institution*. The question analysis algorithm determined that either a *person* or an *institution* are the possible answer types that can potentially "become a university president". Question answering systems can use third party software for processing questions. For example, in *StoQA* I use the NLTK toolkit (Bird et al., 2008) for part of speech tagging and identifying linguistic phrases.[2] Query expansion can be achieved by adding terms to search queries using WordNet (Miller, 1995). A query input to the search engine is constructed from the identified components: words, linguistic phrases, named entities.

In the document and sentence extraction phase, candidate sentences containing target terms are extracted from the documents retrieved by the query. The system can currently retrieve documents from either the Web using the Yahoo search API (Yahoo!, Inc., 2008), or the AQUAINT corpus (Graff, 2002) using the Lucene indexer and search engine (Apache). When using Lucene, I can assign different weights to different types of search term (e.g. less weight to terms than to named entities added to a query) (cf. (Lee et al., 2001)). The candidate sentences are scored according to the number and the type of constituent from the question present in them. Candidate sentences with the higher number of constituents are scored higher.

[2]The impact of identifying linguistic prases on question answering performance is described in Stoyanchev et al. (2008b).

Candidate sentences with named entity and linguistic phrase constituents are scored higher than candidate sentences with single word constituents. The score for each type of constituent is derived empirically and is aimed at increasing the probability for the correct answer appearing in a candidate sentence. The Figure 7.2 shows parts of the two candidate sentences identified and scored in the Document and Sentence Extraction phase. Both candidate sentence examples contain words from the question *president*, a university name *Tufts*, and the correct answer *John DiBiaggio*. The second example also contains a linguistic phrase *"university president"*, which explains the higher score given to the second candidate sentence.

Finally, in the answer extraction phase, the candidate sentences are processed to identify and extract candidate answers, which are presented to the user. I currently have two modules for answer extraction, which can be used separately or together. Candidate sentences can be tagged with named entity information using the Lydia system (Lloyd et al., 2005). The tagged word/phrase matching the target answer type (in the example above *person* or *institute*) most frequently found is chosen as the answer. Candidate sentences can also be tagged with semantic role information using the SRL toolkit from (Punyakanok et al., 2008). In this case, the tagged word/phrase matching the target semantic role most frequently found is chosen as the answer.

7.2.2 Spoken-Input Interactive QA

In a spoken-input QA system, a question is first recognized. To improve speech recognition of a question I simulate an interactive system where the user first specifies a target named entity. The named entity concept is grounded: the user confirms that the named entity is recognized correctly. In the case of continuous misrecognition, a named entity may be spelled. This task has been widely studied in the framework of directory assistance systems (Kellner et al., 1998, among others). A keypad aided spelling correction may be used as a back-off mechanism (Parthasarathy, 2004) where the user uses the phone keypad while spelling the name. In this experiments I am do not address the problem of initial recognition of the named entity.

Figure 7.3 shows the control flow of the simulated system. Resources marked as (0) (grammar of named entities G_{NE} and questions language model LM_Q) are built off-line. G_{NE} is built from the database of named entities and LM_Q is built from the dataset of TREC questions. During runtime the system first requests a user to specify the target named entity (1) and recognizes it (2) with a previously built grammar G_{NE}[3]. Next, the system asks the user to specify a question about the given named entity while it extracts matching documents from the dataset (3) and builds a name-specific language model LM_{DOC}(4). The idea is limiting the language model using the names in consideration. While the name-specific language model built from the documents matching the name, LM_{DOC}, provides the context words, the language model built from questions LM_Q provide the typical characteristics of questions, such as the Wh- words at the sentence initial position. LM_Q is then merged with LM_{DOC} using linear interpolation and generating $LM_{DOC} + LM_Q$(5). The interpolation weight, λ, is kept constant as optimized on a couple of held-out spoken questions.

$$P_{LM}(W) = \lambda \times LM_{DOC}(W) + (1 - \lambda) \times LM_Q(W)$$

The described approach focuses on the improvement in speech recognition of the question by employing interactivity. Once the target named entity is recognized by the system, a target-specific model is built. I use a search engine to extract the documents matching the named entity in the target corpus and use these documents to build the name-specific language model. I hypothesize that these documents are likely to contain the content words of the question resulting in a more relevant model for speech

[3]This step is not evaluated in this project.

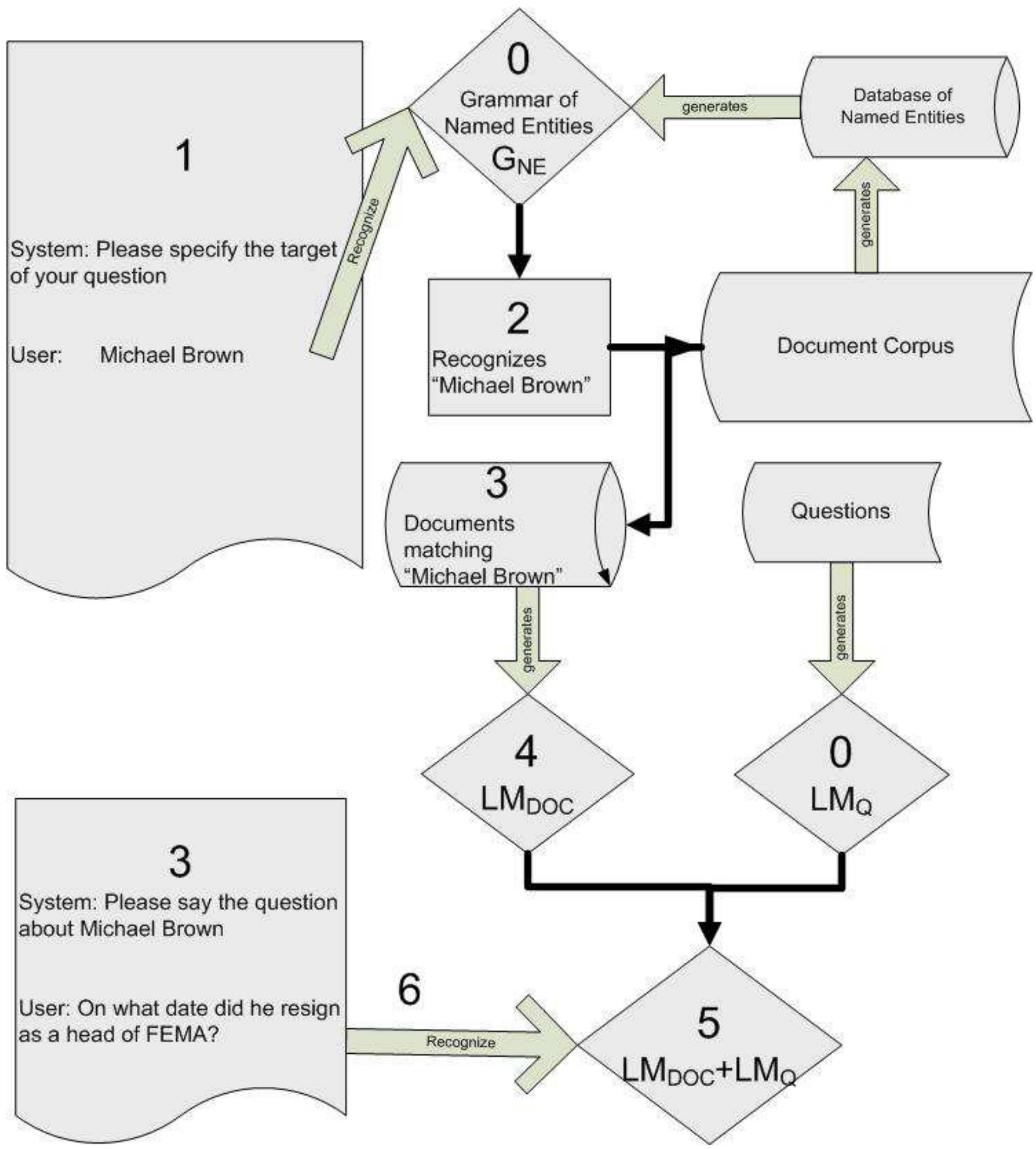

Figure 7.3: Dialog flow example

recognition. For example, for the question *On what date did Michael Brown resign as head of FEMA?*, the words *resign* and *FEMA* may have relatively low probability in a generic model, but higher probability in a model extracted from documents matching *Michael Brown*. The documents are extracted from the AQUAINT corpus and indexed by the Lucene information retrieval engine (Apache). I extract documents matching the string pattern of the target named entity using the Lucene API.

Note that this approach evaluates only name-specific language models. In all experiments I kept the acoustic model fixed. Using this approach while I have a better language model, its size is also smaller than the one obtained using the whole target corpus. This is very important for the efficiency of a real-time recognizer.

7.3 Speech Recognition Experiment

7.3.1 Experimental Approach

In this study I use the TREC annual benchmark evaluation questions targeting the AQUAINT corpus consisting of 3 GB of written news (Dang et al., 2006). The corpus is indexed using the Lucene information retrieval engine (Apache).

Given the question answering architecture described above, how can we generate language models for recognizing spoken queries? I evaluate six approaches to building language models for the spoken-input QA task. These are listed in the Table 7.3. The first four approaches *AQUAINT, Q-2006, Q-2007,* and *AQUAINT-Q2006* use a single open-domain model to recognize all questions and are used in the non-interactive QA scenario (outlined in the Table 7.2). The last two approaches *AQUAINT-perQ* and *AQUAINT-perQ-Q2006* are name-specific models used in the interactive QA scenario. The method of building non-interactive language models is graphically illustrated in Figure 7.4.

Model	type	vocab size	description
Non-interactive models			
AQUAINT	general	3,000	all AQUAINT documents
Q-2006	general	5,012	TREC questions **not** containing test set (total 3713 questions)
Q-2007	general	5,337	TREC questions containing test set (total 4158 questions)
AQUAINT-Q2006	general	6,344	all AQUAINT documents merged with the TREC questions
Interactive models			
AQUAINT-perQ	per target name	7,211	up-to-100 top matches for the target of the question
AQUAINT-perQ-Q2006	per target name	10,210	up-to-100 top matches for the target of the question merged with the TREC questions

Table 7.3: Language models used in the experiment

AQUAINT language model is built with the AQUAINT documents dataset. It is intuitive to build a language model from the dataset used for retrieving candidate documents as the dataset covers all topics expected in the users' questions that can be answered with the information from this dataset. The vocabulary of the language model is pruned. Hence, words with lower frequencies are absent from the language model and will not be recognized if present in a question.

Most sentences in the AQUAINT corpus are statements. Questions recognized in spoken QA task have a different grammatical structure from the sentences in the dataset. Question syntactic structure may not be well represented by the AQUAINT language model and this may cause poor speech recognition. So, the next two approaches (*Q-2006* and *Q-2007*) use language models built from questions.

I build the *Q-2006* model from a set of approximately 4K TREC questions from previous experiments **not** containing the questions in the test set. The experiment

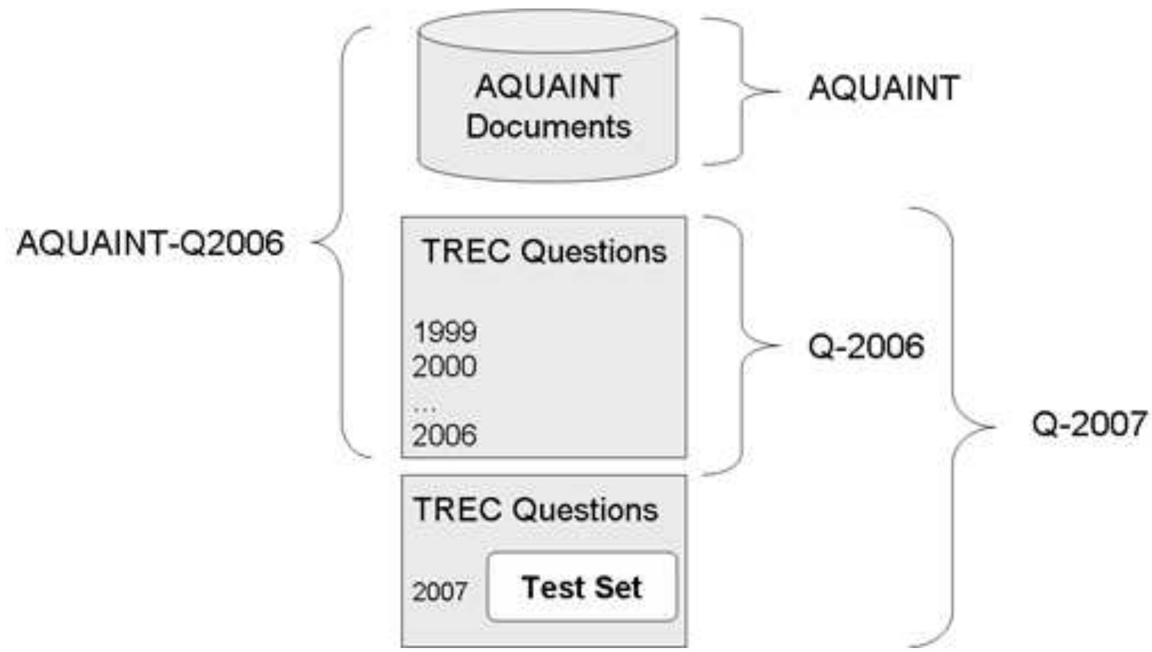

Figure 7.4: Non-interactive language models used in the experiment

with this model simulates a realistic scenario where the users' questions are not known in advance. Given that the set of questions used for building language model is large, it is likely that most of the vocabulary in the test set is covered by the training set used for building the language model. I expect that the speech recognition performance on this dataset will be lower or comparable to the *AQUAINT* dataset.

I build the *Q-2007* model from a set of approximately 4K TREC questions *including* the test questions. The experiment with this model simulates a scenario where the users' questions come from a known larger pool of questions. Generally, in an open-domain QA task, the questions are not known in advance. *Q-2007* model is a "cheeting" model since the set of test questions is a subset of training data used to build this model. I hypothesize that speech recognition of the spoken-input QA will achieve the best performance on this model.

In order to expand the vocabulary of the *Q-2006* model and to enrich *AQUAINT* model with question-specific syntactic constructions, I merge the *Q-2006* and *AQUAINT* models generating *AQUAINT-Q2006*. I expect that the speech recognition on the *AQUAINT-Q2006* will improve over both the *Q-2006* and *AQUAINT* models.

The next two approaches *AQUAINT-perQ* and *AQUAINT-perQ2006* use a question-specific model to recognize each question. I build *AQUAINT-perQ* model from up to 100 documents extracted from the AQUAINT dataset matching the previously specified name. I hypothesize that the recognition performance on this model will improve over the baseline *AQUAINT* model. I build *AQUAINT-perQ-Q2006* model by merging *AQUAINT-perQ* and *Q-2006* models. I hypothesize that the recognition performance on this model will achieve a further improvement. Although the experiments described in this chapter were run off-line, generation of a question-specific language model can be efficiently implemented for an on-line system.

7.3.2 Questions Datasets

As my test set I have selected 40 questions from the TREC 2007 evaluations. For 18 of the selected questions the target is a person, for 17 of the questions the target is an organization, and 5 of the questions have another type of target.

Original TREC question	How many times has **Limbaugh** been married?
Target NE	Rush Limbaugh
Modified with NE (*WithNE*)	How many times has **Rush Limbaugh** been married?
Modified without NE (*NoNE*)	How many times has **he** been married?

Table 7.4: Example of a question in the test set

The questions are modified for my experiments. In the *WithNE* set, all questions are modified to contain the target named entity. That is, if the original question contains a pronoun referring to the target named entity, it is replaced with an appropriate form of the target. In *NoNE*, all questions are modified to *not* contain the named entity by replacing it with an appropriate pronoun. Table 7.4 illustrates how the questions are modified for the experiment.

The 40 questions with resolved and 40 questions with unresolved named entities are read and recorded by three subjects. One of the subjects was the the author. The other two subjects were native english speakers with college education not involved in this project.

I compare recognition of the test questions using target-specific language models with recognition using a generic language model. All models in this experiment are built using the the SRILM language modeling toolkit (Stolcke, 2002). The speech recognition experiments are performed using SRI's Dynaspeek^{TM}speech recognition system (Franco et al., 2002).

7.3.3 Experimental Results

In the Figure 7.5 I report the average word error rate over 40 questions on both *WithNE* and *NoNE* sets using each of the models described above.

The *AQUAINT* model has the highest word error rate of 58.36% on the *WithNE* and 46.64% on the *NoNE* sets. Although the AQUAINT corpus has large vocabulary coverage, the form of the questions differs from the form of the sentences in the corpus (such as sentences starting with Wh- words).

The *Q-2006* model is likely to lack the target named entities of the test set. Nevertheless, *Q-2006* model has a lower error rate comparing to *AQUAINT* of 45.65% on the *WithNE* and 32.13% on the *NoNE* set.

As expected, the *Q-2007* model achieves the lowest word error rate of 19.77% on the *WithNE* and 17.27% on the *NoNE* sets. Notice that although the *NoNE* test set does not contain names, the word error rate using the *Q-2006* model is almost twice as high as the word error rate using *Q-2007* model. This difference can be caused by the lower recognition of content and function words (other than names) on *Q-2006*.

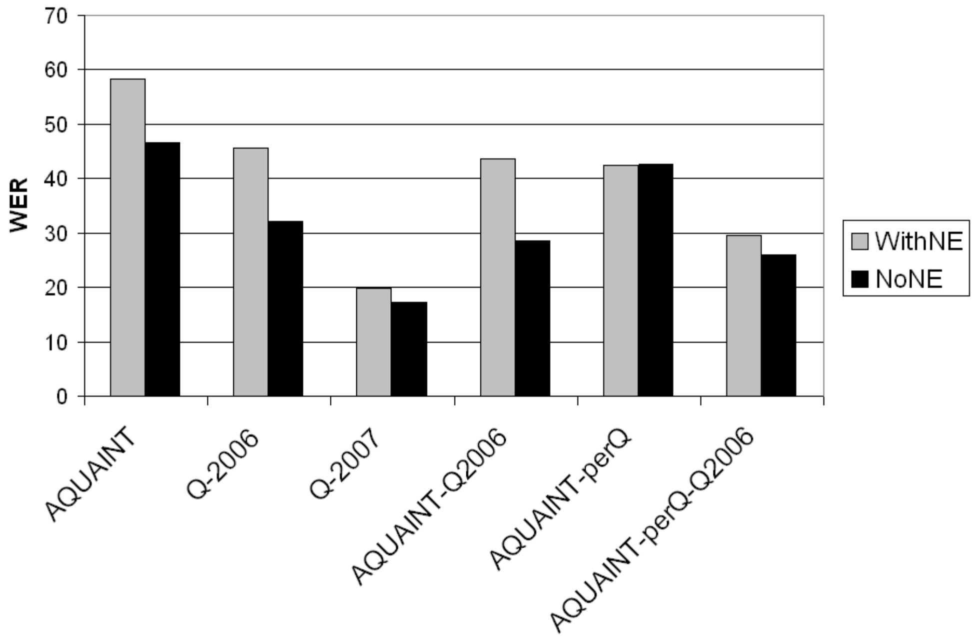

Figure 7.5: WER of recognizing spoken questions

This shows the importance of the content words associated with the target names.

AQUAINT-Q2006 model reduces the error rate compared to *AQUAINT* by 25% on the *WithNE* and 40% relatively on the *NoNE* set. Its higher word error rate reduction on the the *NoNE* sent than on the *WithNE* set suggests that the recognition improvement is due to the better recognition of content words and not named entities.

Next, I report the results on a per-question *AQUAINT-perQ* model. Surprisingly, it has a high WER of 42.5% on the *WithNE* and 42.6% on the *NoNE* set. *AQUAINT-perQ*, similarly to *AQUAINT* is built on statements and not questions. The high error rate on both sets suggests the importance of a language model reflecting question sentence structure.

My final model *AQUAINT-perQ-Q2006* is a merger of the *AQUAINT* model with the *Q-2006* model. This model achieves the lowest WER among all tested models (except the "cheating" Q-2007 model) of 32.4% on the *WithNE* and 28.7% on the

NoNE set. This is a relative reduction of 32.2% on the *WithNE* set compared to the best generic model performance.

Model	% err reduction	% err reduction
AQUAINT	49.4	44.2
Q-2006	35.3	19.0
AQUAINT-Q2006	32.2	8.7
AQUAINT-perQ	30.5	38.9

Table 7.5: Relative error reduction for the AQUAINT-perQ-Q2006 model

Table 7.5 reports the relative word error rate reduction for the test model *AQUAINT-perQ-Q2006* from each other model. Note that, in addition to the dramatic reduction in word error rate, the ratio of missed named entity recognitions is halved, coming down to levels which can be obtained using the cheating experiment.

There are also different word error rates for the three speakers. Speakers 1 and 2 have higher word error rate on the *WithNE* set than on the *NoNE* set for all the models; however, speaker 3 achieves higher word error rate on the *NoNE* set for the *AQUAINT-perQ* model. It is possible that speaker 3 was very clear in pronouncing the target named entities and was able to achieve lower word error rate on the models that contain target named entities.[4]

7.4 Discussion

In this chapter I presented an approach for improving speech recognition of spoken questions for the open-domain spoken-input question answering task. In this approach I adapted the language model to the name in a question and generated a language model specific to each question. The results show an improvement in

[4]I did not compare means of the WER between the speakers with ANOVA because the distribution of WER is not normal

speech recognition performance using a language model adapted to the name. The best speech recognition result was achieved using a model built from a combination of the documents containing the question-specific name and the 4000 questions datasets.

Speech recognition using the models built only from documents, even when these documents contain the name in the question, had a high word error rate. Speech recognition using the models built only from questions also had a high word error rate. The result of this study points to the importance of including the data with the question-specific vocabulary as well as the data with the question-specific syntax in the language model.

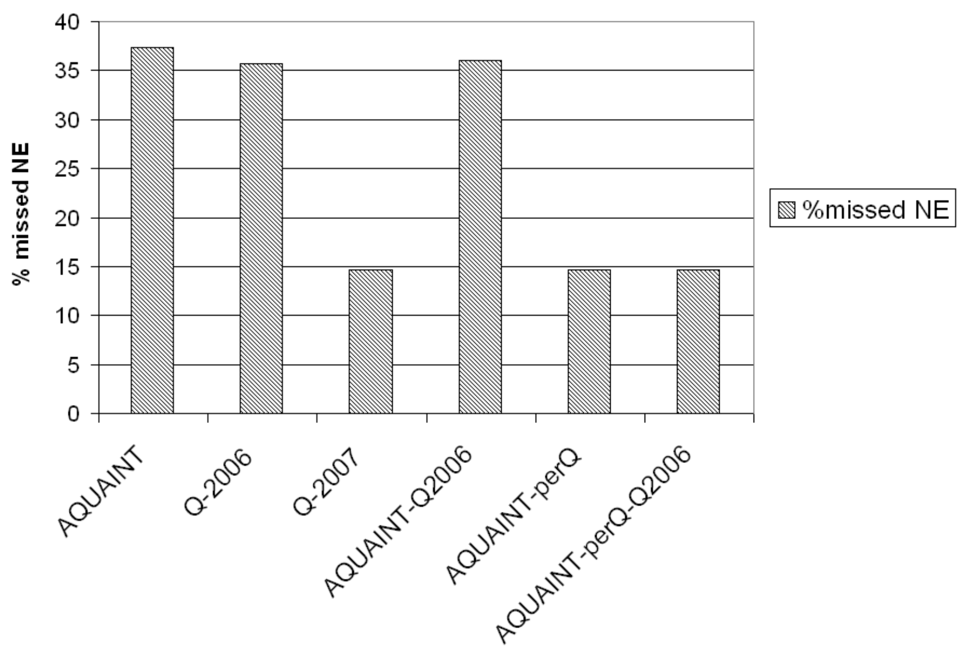

Figure 7.6: % of missed names during speech recognition

In these experiments I assumed that the name in the question is recognized correctly. However, in a more realistic scenario the system does not know the named entity in advance. The word error rate on the dataset containing named entities

(*WithNE*) is consistently higher for all language models. Figure 7.6 shows the percent of names missed during recognition of the *WithNE* set. With the non-interactive and "non-cheating" approaches (*AQUAINT, Q-2006,* and *AQUAINT-Q2006*) ASR misses over 35% of names. By contrast, in the "cheating" *Q-2007* model and both of the interactive approaches *AQAINT-perQ* and *AQAINT-perQ-Q2006*, ASR misses only 14% of names. Recognition using these language models achieves such a low word error rate because the target named entity is present in the data used to build the model. An important future work for interactive spoken-input QA approach includes work on recognition of the name in the first stage of the interaction. A possible method involves automatic extraction of named entities from the AQUAINT corpus and building a grammar from these named entities. This list, of course, can be large and may contain similar and easily confusable words (cognates)[5].

To resolve this problem, the system can first ask the user to specify the name in the question and use the grammar of named entities to recognize it. Then the system can build N per-question language models using the top N recognition hypotheses. Once the user specifies the question, the system can recognize the question with all N models. The hypothesis with the highest ASR score can be used as the final recognition hypothesis. Several proposed approaches to solving the name recognition problem in a spoken-input question are outlined in proposal (Stoyanchev, 2009).

[5]Goldwater et al. (2008) idendify cognates as one of the top reasons for recognition errors.

Chapter 8

Conclusions and Future Work

In this dissertation, I explore adaptation in spoken dialog. Adaptation is exhibited by the convergence of language use and interactive behavior of two agents in a dialog. I hypothesize that dialog participants maintain a *conversation model* and use it to adapt both to specific dialog partners and to recent dialog behaviors. In my work I have analyzed how adaptation may be used to improve the performance of automatic spoken dialog systems.

I address the question of adaptation from three different perspectives. First, I analyze adaptation in a human-human dialog corpus, and show that there is lexical and syntactic adaptation both within and between dialogs. Second, in an empirical study I show that users adapt to lexical and syntactic choices reflected in system prompts, and that this adaptation has an effect on dialog system performance. Third, I show that automatic speech recognition performance of a dialog system can be improved by adapting the system's language model to the concepts and topic of the user. In the following sections I summarize the findings presented in this dissertation, describe implications for dialog system development, and outline potential directions for future research.

8.1 Summary of Findings

8.1.1 Measuring Adaptation

In this study I compared adaptation due to the partner and due to recency in a human-human spoken dialog corpus. I devised a new method to measure adaptation *between* dialogs. I used this method to examine lexical, syntactic and perspective adaptation in Maptask dialogs (Anderson et al., 1991). I measured adaptation to the most recent conversation partner and to the specific conversation partner.

For lexical features overall, the difference between the prevalence of adaptation due to dialog partner and adaptation due to recency was not significant. I found that primed lexical stem features are approximately 2.7 times more likely to appear in later dialogs than they would be predicted to appear by chance. Primed bigram features are approximately 3 times more likely to appear in later dialogs than they would be predicted to appear by chance.

For syntactic features, on average, adaptation to the most recent partner is stronger than to the specific partner. Primed syntactic features are on average 2.9 times more likely to appear in the very next dialog independent of the partner and 2.7 times in a later dialog with the same partner compared to their appearance by chance.

I also used my new adaptation measure to identify features that tend to exhibit partner adaptation and those that tend to exhibit recency adaptation. My results suggest that the semantic category of a lexical feature may affect whether the feature is likely to exhibit partner adaptation or recency adaptation. For example, words indicating *direction*, such as *across* and *through*, are more likely to exhibit partner adaptation than recency adaptation. My results show that sentence structures, such as complex sentences (S→S S), are more likely to exhibit partner adaptation than recency adaptation, while most noun phrase constructions exhibit recency adaptation, particularly noun phrases with possessives (*his, her, mine, yours, etc.*) or negations.

One interesting direction for future work on adaptation is to measure the correlation between task success and adaptation of the users in spoken dialog (Schober and Brennan, 2003). This can be done by correlating dialog length, number of errors, or successful completion of a task.

8.1.2 Directive Adaptation

In this set of studies I analyzed user adaptation to the form of system prompts in human-computer dialog. Previous laboratory studies of adaptation in human-computer dialog have shown lexical and syntactic adaptation. In my work I extended these findings using a live spoken dialog system with real users.

I used a modified version of the deployed *Let's Go!* system for these studies. I examined adaptation to the use of verbs, prepositions, verb forms, and forms of task-related concepts. I found that presence of a verb in a system prompt increases the probability of a verb being used in the user's response also. I found that users indeed are more likely to use the same verb form and concept form as the system. I found a trend in adaptation to the use of prepositions, however the difference was not statistically significant.

I also compared user adaptation in dialogs with *adapting system condition* (in which the system used the same concept form as a user) and *non-adapting system condition* (in which the system used a different concept form than the user). I found that in the adapting system condition, the user is likely to repeat the same concept form, while in the non-adapting system condition the user is likely to change the concept form. However, contrary to the findings of previous work that the user changes his/her concept form to the form used by the system, users in the *non-adapting system condition* changed their concept form randomly to any other concept form. The adapting system condition led to predictable user behavior in terms of choice of concept realization. The non-adapting system condition led to unpredictable user behavior where concept forms could not be accurately predicted.

Today, most dialog systems do not model user or system adaptation. Perhaps this adaptation is of less importance to *limited input* dialog systems. However, *flexible input* dialog systems can benefit from this adaptation phenomena through the use of (a) directive prompts that guide users to using language that can be better understood by the system, and (b) responsive adaptation that reinforces user adaptation in confirmation prompts.

8.1.3 Responsive Adaptation

In this set of studies I evaluated the impact of responsive adaptation on the performance of the speech recognition component of a dialog system. The scientific motivation for this study was the idea that human speech production and comprehension are tightly coupled and consequently a dialog system's expectations about the content of a speaker's utterance can help it in utterance interpretation (Pickering and Garrod, 2007). The engineering motivation for this research was the evidence that smaller and more targeted language models tend to have better performance for speech recognizers.

In one experiment I evaluated prediction of the task-related concepts in user's utterance. The results of my experiment suggest that prosodic and dialog history features are useful for predicting task-related concepts in a user utterances, and that adaptation of the speech recognizer's language model to the predicted concepts can lead to small but significant improvements in speech recognition accuracy. In a similar experiment on a spoken question answering system, I showed that language model adaptation to the name in a user's question can lead to significant improvements in speech recognition accuracy.

My experiments show that adapting language models to contextually appropriate lexical forms leads to improvement in speech recognition performance. This has implications for other types of open-domain recognition tasks. For example, recognition of news reports can use detection of topic shifts to adapt language models.

In this work I focused on the prediction of semantic content in a user's utterance. However, my method did not address another major cause of speech recognition errors, speech disfluencies (Goldwater et al., 2008). Disfluencies include filler words such as *um* and *uh*, restarts, and hesitations. Disfluencies are prevalent in speech directed at telephone-based spoken dialog systems. In future work I would like to incorporating disfluency and prosodic/acoustic models in spoken dialog systems. I hypothesize that disfluency models are speaker-dependent and require the system to adapt to a specific user.

8.2 Implications

My work has implications for the design of dialog systems. To model and take advantage of adaptation, a dialog system architecture must: 1) facilitate resource sharing between components; 2) use flexible output components; and 3) use flexible input components.

8.2.1 Resource Sharing between Components

In my responsive adaptation experiment (see Chapter 6) I showed that speech recognition accuracy can be improved by adapting the language model according to predicted topics or concepts in a user's utterance. In my directive adaptation experiments (see Chapter 5) I showed that words and syntax used in system prompts affect users' lexical and syntactic choices, and that this can affect system performance. These findings suggest the need for information and resource sharing between system components implementing different system functionality: speech recognition and language understanding (ASR, NLU), dialog management (DM), and natural language generation (NLG) (Kempson et al., 2009).

Given that users adapt to the system's lexical and syntactic choices, the ASR and NLU components can benefit from dynamically adjusting probabilities of the words and grammatical structures recently used by the system and generated by the NLG component. For example, in one of my experiments I showed that if a system prompt uses a verb, the user is more likely to also use a verb in their utterance. A system may take advantage of this information and adjust probabilities in the language model of the ASR: if the system prompt uses a verb, the rules in the ASR grammar containing a verb may get more probability mass while the rules without a verb may get less.

I also found that adaptive system behavior leads to more predictable user behavior. I hypothesize that adaptation of NLG to the users' lexicon and grammar may have a positive effect on speech recognition accuracy. For example, a dialog system may mimic the structure of a prepositional phrase attachment in a user's utterance (e.g. *taking four o'clock bus* vs. *taking a bus at four*) or adapt to the verb choice (e.g. *leaving* vs. *departing*), or adapt to the form of a concept (e.g. *Madison and Fifth* vs. *Fifth and Madison*). The system's choice to adapt to the user's lexical and syntactic choices may lead to a higher likelihood of the user repeating the structure in later utterances and provide more information for adjusting the ASR's language model, which can then lead to improved task success.

My proposed adaptive dialog system architecture is shown in Figure 8.1. The system stores shared histories for itself and the user: their syntactic choices, lexical choice, dialog acts, topics, etc. The history is updated dynamically when the user's input is recognized by the ASR and parsed by the NLU, or when a prompt is generated by the NLG, or when an action is taken by the DM.

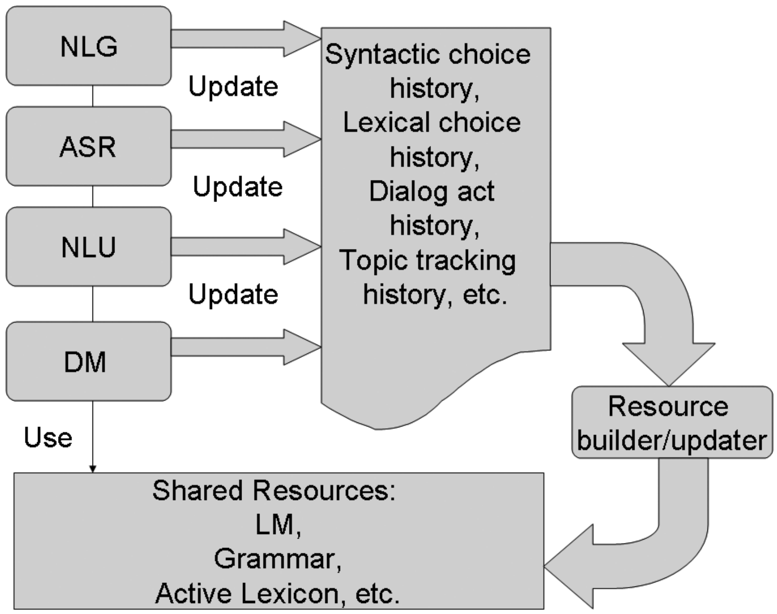

Figure 8.1: Architecture of an adaptive resource sharing dialog system

8.2.2 Flexible Output Components

The proposed resource sharing and adaptation place a flexibility requirement on the dialog system NLG components. If NLG is fixed (e.g. the same prompt formulation is used in all situations), it can not be adapted to the user or to the dialog context. On the other hand, a flexible (or trainable) NLG component allows for dynamic prompt generation.

Prompts in most modern task-oriented systems are designed using templates, i.e. prompts hand-written by a voice user interface (VUI) designers. This allows VUI designers to manually create an alignment between ASR and NLG. For example, if according to the existing data for an application, the sentence *schedule an event* that has a higher rate of recognition than the alternative *add an event*, a VUI designer may use *schedule* instead of *add* in prompts, e.g. *Would you like to schedule an event?* instead of *Would you like to add an event?*. However, manual VUI design is tractable only in systems where the number of prompts is relatively small. Even in small systems, the potential for manual alignment between user utterances and a system prompts is limited. Although VUI designers may be aware of certain problematic cases in their dialog systems (such as the choice of *add* vs. *schedule*), it would be difficult to manually exploit all potential lexical and syntactic alignment possibilities in a dialog system.

A trainable NLG automatically produces system prompts while taking into account user-specific or dialog-specific information. This makes prompts dynamic and flexible. With a trainable NLG, the same prompt may be realized differently based on the speech recognizer's estimated performance, dialog length, or the user's approximated age or gender (van der Sluis and Mellish, 2009). I believe that even limited domain dialog systems, such as *Let's Go!*, can benefit from a trainable NLG component.

8.2.3 Flexible Input Components

Humans are able to recognize speech even with high levels of background noise. Context helps us immensely in speech recognition. People recognize more easily the words that are predictable from context than the words that are not predictable from context. I hypothesize that input components in dialog systems can be improved by giving the same context, with the use of *dynamic language models*. For example, a dynamic language model is updated throughout a dialog based on a user model, the dialog context, user's vocabulary and prosody.

I have shown in this thesis that language model adaptation to the topic of a user's utterance and expected concepts in a user's utterance can improve speech recognition accuracy. In the experiments in this thesis I used dialog act history and prosodic information for language model adaptation. This is a first step towards dynamic language modeling in a dialog system.

Dialog system architectures must facilitate information passing and resource sharing for the ASR to be able to build or adapt dynamic language models on the fly or for the NLU to dynamically adapt its grammar. My proposed architecture (Figure 8.1) allows use of feature combinations from all system components (NLG, NLU, ASR, and DM) to dynamically adapt the input system components.

8.3 Future Directions of Research

8.3.1 Evaluate Directive Adaptation

I have shown that users of dialog systems are affected by the systems' lexical and syntactic choices. My future work includes evaluation of how adaptation can be utilized by the system and whether it can lead to improvements in various measures of dialog system performance.

The *resource sharing* architecture I propose in the previous section allows to evaluate the effect of different features on dialog adaptation and system performance. In this thesis I have shown that dialog history and utterance prosody are useful features in content prediction for ASR adaptation. In future work I would like to evaluate how this effect can be used for improving system performance.

I hypothesize that the *directive power* of system prompts will be more significant in dialog systems with more complex domains, such as tutoring system or a virtual assistants. These systems have a larger vocabulary and topic space than a bus information or an airline ticket system. Hence, they have more opportunities for contextual adaptation.

8.3.2 User Categorization

In this thesis all dialog system users were treated as the same. However, users have different personalities and backgrounds. They have different characteristics in communication with a dialog system. For example, some users may be more verbose than others, some users may be more polite than others. While specifying information, some users may prefer to specify all information in a single utterance, e.g. *When is the next 28X from Downtown to the Airport?* while others may prefer to specify it step by step in consecutive utterances, e.g. first specify departure location: *Leaving form Downtown*, then specify destination: *Going to Airport*, and finally bus route: *28X*. User behavior following dialog system errors may also differ. In some situations, users may be more successful with correction when the system confirmation prompt specifies all information: *From Downtown to the Airport, is this correct?*, while in other cases users may prefer to have each piece of information confirmed separately. Knowledge about the user, interaction behavior, or preferences may be used in NLG, ASR, and NLU components to improve system performance. I hypothesize that users may be categorized according to their communication patterns (Doddington et al., 1998). This information may then be used by the system as a feature for 1) predicting user utterance content and possible improvement of ASR performance, 2) choosing realization of a prompt, and 3) choosing a dialog move.

8.3.3 Adaptation and Relearning

A need for changes in dialog system functionality is a common scenario. Changes may involve the wording of system prompts, topic order, addition of a new topics. Functionality changes aim at improving system performance, however the users *learn* or get *adapted* to a particular spoken interface, just as they learn to use a particular graphical interface.

One direction for future work is to determine how to make dialog system changes less disruptive for the user through adaptation. I would like to study which properties of a change in a system cause disruption and confusion and how a system can guide a user to adapt to the changes in the interface.

8.4 Take-Home Message

What is the future of dialog systems? On the one hand, we have *limited input* systems that understand key words and phrases requiring users to learn. On the other hand we have *flexible input* systems allowing users to speak naturally. One possibility is that people will become proficient at using *limited input* dialog systems and there will be no need for *natural* verbal communication. However, I would argue that given a reasonable system performance, *natural* verbal communication may be preferable for most users. My experimental results suggest that:

- Adaptation occurs within and between dialogs.
- Responsible adaptation in dialog system can lead to improved system performance.
- Directive adaptation can lead to changes in user behavior, which can also lead to improved system performance.
- In order for the potential benefits for adaptation to be realized in dialog systems substantial changes to the architecture are needed to support more information sharing. However, even light-weight adaptation has impact.
- This is just the beginning. Much more research is needed on how adaptation works and how it can impact different measures of dialog success.

Bibliography

H. Ai, A. Raux, D. Bohus, M. Eskenazi, and D. Litman. Comparing spoken dialog corpora collected with recruited subjects versus real users. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialog*, 2007.

A. Anderson et al. The HCRC map task corpus. *Language and Speech*, 34:351–366, 1991.

Apache. Lucene search engine. http://lucene.apache.org/.

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.

L. Bell, J. Boye, J. Gustafson, and M. Wiren. Modality convergence in a multimodal dialogue system. In *Proceedings of 4th Workshop on the Semantics and Pragmatics of Dialogue (GOTALOG)*, 2000.

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

BeVocal. http://cafe.bevocal.com.

S. Bird, E. Loper, and E. Klein. Natural Language ToolKit (NLTK), 2008. http://nltk.org/.

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.

D. Bohus, B. Langner, A. Raux, A. Black, M. Eskenazi, and A. Rudnicky. Online supervised learning of non-understanding recovery policies. In *Proceedings of SLT*, 2006.

S. Brennan. Conversations with and through computers. *User Modeling and User-Adapted Interaction*, pages 67–86, 1991.

S. Brennan. Lexical entrainment in spontaneous dialog. In *Proceedings of the International Symposium on Spoken Dialog*, 1996.

S. Brennan. The vocabulary problem in spoken dialogue systems. *Automated spoken dialog systems (unpublished manuscript)*, 1998.

S. Brennan and H. Clark. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, 22(6):1482–1493, 1996.

S. Brennan and E. Hulteen. Interaction and feedback in a spoken language system: A theoretical framework. *Knowledge-Based Systems*, 8:143–151, 1995.

S. Brennan and J.O. Ohaeri. Effects of message style on user's attribution toward agents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1994.

P. Brown and G. Dell. Adapting production to comprehension: The explicit mention of instruments. *Cognitive Psychology*, 19:441–472, 1987.

T. Chartrand and J. Bargh. The chameleon effect: The perception-behavior link and social interaction. *Journal of Personality and Social Psychology*, 76:893–910, 1999.

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.

K. Church. Empirical estimates of adaptation: the chance of two noriegas is closer to p/2 than p2. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2000.

H. Clark and E. Schaefer. Contributing to discourse. *Cognitive Science*, 13(6):259–294, 1989.

H. T. Dang, J. Lin, and D. Kelly. Overview of the text retrieval conference (trec) 2006 question answering track. In *NIST Special Publication: SP 500-272*, 2006.

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.

G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1998.

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. Mori. Stochastic finite state automata language model triggered by dialogue states. In *Proceedings of Eurospeech*, 2001.

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.

H. Franco, J. Zheng, J. Butzberger, F. Cesari, M. Frandse, J. Arnold, V. Ramana, R. Gadde, A. Stolcke, and V. Abrash. DynaSpeak: SRI's scalable speech recognizer for embedded and mobile systems. In *Proceedings of the Human Language Technology Conference (HLT)*, San Diego, CA, 2002.

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 the International Conference on Spoken Language Processing (ICSLP)*, 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 G. Doherty. Conversation, coordination and convention: An empirical investigation of how groups establish linguistic conventions. *Cognition*, 53:181–215, 1994.

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

S. Goldwater, D. Jurafsky, and C. Manning. Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase ASR error rates. In *Proceedings of the Joint Meeting of the Association for Computational Linguistics and Human Language Technology Conference (ACL/HLT)*, 2008.

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.

D. Graff. The AQUAINT corpus of English news text. Technical report, Linguistic Data Consortium, Philadelphia, PA, USA, 2002.

P. Grice. Syntactic coordination in dialogue. *Radical Pragmatics, Academic Press*, pages 183–198, 1981.

H. Guo and A. Stent. Trainable adaptable multimedia presentation generation. In *Proceedings of the International Conference on Multimodal Interfaces*, 2005.

J. Gustafson, A. Larsson, R. Carlson, and K. Hellman. How do system questions influence lexical choices in user answers? In *Proceedings of Eurospeech*, 1997.

R. J. Hartsuiker, S. Bernolet, S. Schoonbaert, S. Speybroeck, and Dieter Vanderelst. Syntactic priming persists while the lexical boost decays: Evidence from written and spoken dialogue. *Journal of Memory and Language*, 2007.

J. Henderson, O. Lemon, and K. Georgila. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34:487–511, 2006.

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.

W. Horton and R. Gerrig. Speakers' experiences and audience design: knowing when and knowing how to adjust utterances to addressees. *Journal of Memory and Language*, 47:589–606, 2002.

D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky. PocketSphinx: 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.

A. Kellner, B. Rueber, and H. Schramm. Strategies for name recognition in automatic directory assistance systems. In *Proceedings of the 4th IEEE Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA)*, Torino, Italy, 1998.

R. Kempson, E. Gregoromichelaki, and Y. Sato. Incrementality, speaker-hearer switching and the disambiguation challenge. In *Proceedings of the EACL Workshop on the Semantic Representation of Spoken Language*, 2009.

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.

K. Komatani. User modeling in spoken dialog systems to generate flexible guidelines. In *Proceedings of the Conference on User Modeling and User-Adapted Interaction (UMUAI)*, 2005.

K. Komatani, Y. Fukubayashi, T Ogata, and H. Okuno. Introducing utterance verification in spoken dialogue system to improve dynamic help generation for novice users. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialog*, 2007.

T. Kraljic and S. Brennan. Prosodic disambiguation of syntactic structure: For the speaker or for the addressee? *Cognitive Psychology*, 50:194–231, 2005.

R. Kraut, S. Fussell, and J. Siegel. Visual information as a conversational resource in collaborative physical tasks. *Human-Computer Interaction*, 18(1–2):13–49, 2003.

I. Kruijff-Korbayova and O. Kukina. The effect of dialogue system output style variation on users' evaluation judgments and input style. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialog*, 2008.

G. G. Lee et al. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In *Proceedings of Text REtrieval Conference (TREC)*, 2001.

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.

O. Lemon, K. Georgila, and J. Henderson. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: The Talk TownInfo evaluation. In *prodeedings SLT*, 2006.

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

L. Lloyd, D. Kechagias, and S. Skiena. Lydia: A system for large-scale news analysis. In *Proceedings of String Processing and Information Retrieval*, pages 161–166, 2005.

C. Lockridge and S. Brennan. Addressees' needs influence speakers' early syntactic choices. *Psychonomics Bulletin and Review*, 9:550–557, 2002.

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

George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 1995.

A. Nenkova, A. Gravano, and J. Hirschberg. High frequency word entrainment in spoken dialogue. In *Proceedings of the Joint Meeting of the Association for Computational Linguistics and Human Language Technology Conference (ACL/HLT)*, 2008.

S. Parthasarathy. Experiments in keypad-aided spelling recognition. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Montreal, 2004.

J. Pearson, J. Hu, H. Branigan, M. Pickering, and C. Nass. Adaptive language behavior in HCI: How expectations and beliefs about a system affect users' word choice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1177 – 1180, 2006.

M. Pickering and S. Garrod. Toward a mechanistic psychology of dialog. *Behavioral and Brain Sciences*, 27:169–190, 2004.

M. Pickering and S. Garrod. Do people use language production to make predictions during comprehension? *Trends in Cognitive Sciences*, 11:105–110, 2007.

M. Pickering, H. Branigan, A. Cleland, and A. Stewart. Activation of syntactic priming during language production. *Journal of Psycholinguistic Research*, 29(2): 205–216, 2000.

V. Punyakanok, D. Roth, and W. Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2), 2008.

M. Purver and R. Kempson. Context-based incremental generation for dialogue. In *Proceedings of the International Natural Language Generation Conference (INLG)*, 2004.

L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE Acoustics, Speech, and Signal Processing Newsletter*, pages 4–15, January 1986.

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.

E. Reitter and F. Keller. Against sequence priming: Evidence from constituents and distituents in corpus data. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, 2007.

E. Reitter and J. Moore. Predicting success in dialogue. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2007.

E. Reitter, F. Keller, and J. Moore. Computational modelling of structural priming in dialogue. In *Proceedings of the Joint Human Language Technology Conference and Meeting of the North American Association for Computational Linguistics (HLT/NAACL)*, 2006a.

E. Reitter, J. Moore, and F. Keller. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *Proceedings of CogSci*, 2006b.

G. Riccardi and S. Bangalore. Automatic acquisition of phrase grammars for stochastic language modeling. In *Proceedings of the Sixth Workshop on Very Large Corpora*, 1998.

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.

M. Rotaru and D. Litman. Dependencies between student state and speech recognition problems in spoken tutoring dialogues. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2006.

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.

M. F. Schober and S. E. Brennan. *Handbook of discourse processes Processing*, pages 123–164. Lawrence Erlbaum, Hillsdale, NJ, 2003.

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 and Technology*, pages 103–111, 2003.

J. Shin, S. Narayanan, L. Gerber, A. Kzemzadeh, 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.

G. Skantze. Coordination of referring expressions in multimodal human-computer dialogue. 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.

H. Soltau. *Compensating Hyperarticulation for Automatic Speech Recognition*. PhD thesis, Karlsruhe University, 2005.

S. Stenchikova and A. Stent. Measuring adaptation between dialogs. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialog*, 2007.

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, R. Prasad, and M. Walker. Trainable sentence planning for complex information presentations in spoken dialog systems. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, 2004.

A. Stent, S. Stenchikova, and M. Marge. Dialog systems for surveys: The Rate-a-Course system. In *Proceedings of the IEEE/ACL Spoken Language Technology Workshop (SLT)*, 2006.

A. Stolcke. SRILM – An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Denver, CO, September 2002.

S. Stoyanchev. Speech Recognition of Names in Spoken Queries, 2009. Unpublishd manuscript.

S. Stoyanchev and A. Stent. An analysis of user behavior following speech recognition errors in the communicator corpus of human-computer spoken dialog. In *Proceedings of the Corpus Linguistics Conference*, 2009a.

S. Stoyanchev and A. Stent. Predicting concept types in user corrections in dialog. In *Proceedings of the EACL Workshop on the Semantic Representation of Spoken Language*, 2009b.

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)*, 2009c.

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)*, 2008a.

S. Stoyanchev, Y. C. Song, and W. Lahti. Exact phrases in information retrieval for question answering. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2008b.

C. Thompson, M. Goker, and P. Langley. A personalized system for conversational recommendations. *Artificial Intelligence*, 21:393–428, 2004.

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*, 2005.

I. van der Sluis and C. Mellish. Towards empirical evaluation of affective tactical nlg. In *Proceedings of the 12th European Workshop on Natural Language Generation*, 2009.

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)*, Denver, CO, September 2002.

M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28:811–840, 2004.

A. Ward and D. Litman. Automatically measuring lexical and acoustic/prosodic convergence in tutorial dialog corpora. In *Speech and Language Technology in Education*, 2007a.

A. Ward and D. Litman. Dialog convergence and learning. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED)*, 2007b.

A. Ward and D. Litman. Semantic cohesion and learning. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS)*, 2008.

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.

J. D. Williams, P. Poupart, and S. Young. Partially observable Markov decision processes with continuous observations for dialogue management. *Computer Speech and Language*, 21:393–422, 2007.

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.

Yahoo!, Inc. Yahoo! search API, 2008. http://developer.yahoo.com/search/.

S. Young. Detecting misrecognitions and out-of-vocabulary words. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1994.

H. Yu, T. Tomokiyo, Z. Wang, and A. Waibel. New developments in automatic meeting transcription. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.

V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. JUPITER: a telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):85–96, 2000.