Harvesting and Summarizing User-Generated Content for Advanced Speech-Based Human-Computer Interaction

by

Jingjing Liu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2012

Signature of Author ……………………………………………………………………...
Department of Electrical Engineering and Computer Science
December 29, 2011

Certified by ……………………………………………………………......................
Stephanie Seneff
Senior Research Scientist
Thesis Supervisor

Certified by ……………………………………………………………......................
Victor Zue
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ……………………………………………………………......................
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Harvesting and Summarizing User-Generated Content for Advanced Speech-Based Human-Computer Interaction

by

Jingjing Liu

Submitted to the Department of Electrical Engineering and Computer Science
on December 29, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

There have been many assistant applications on mobile devices, which could help people obtain rich Web content such as user-generated data (e.g., reviews, posts, blogs, and tweets). However, online communities and social networks are expanding rapidly and it is impossible for people to browse and digest all the information via simple search interface. To help users obtain information more efficiently, both the interface for data access and the information representation need to be improved. An intuitive and personalized interface, such as a dialogue system, could be an ideal assistant, which engages a user in a continuous dialogue to garner the user's interest and capture the user's intent, and assists the user via speech-navigated interactions. In addition, there is a great need for a type of application that can harvest data from the Web, summarize the information in a concise manner, and present it in an aggregated yet natural way such as direct human dialogue. This thesis, therefore, aims to conduct research on a universal framework for developing speech-based interface that can aggregate user-generated Web content and present the summarized information via speech-based human-computer interaction. To accomplish this goal, several challenges must be met. Firstly, how to interpret users' intention from their spoken input correctly? Secondly, how to interpret the semantics and sentiment of user-generated data and aggregate them into structured yet concise summaries? Lastly, how to develop a dialogue modeling mechanism to handle discourse and present the highlighted information via natural language? This thesis explores plausible approaches to tackle these challenges. We will explore a lexicon modeling approach for semantic tagging to improve spoken language understanding and query interpretation. We will investigate a parse-and-paraphrase paradigm and a sentiment scoring mechanism for information extraction from unstructured user-generated data. We will also explore sentiment-involved dialogue modeling and corpus-based language generation approaches for dialogue and discourse. Multilingual prototype systems in multiple domains have been implemented for demonstration.

Thesis supervisor: Stephanie Seneff
Title: Senior Research Scientist, Department of Electrical Engineering and Computer Science

Thesis supervisor: Victor Zue
Title: Professor, Department of Electrical Engineering and Computer Science

Acknowledgments

I would like to express my sincere gratitude to my advisors, Victor Zue and Stephanie Seneff. Victor inspired me to pursue a PhD at MIT and I am thankful for the many opportunities he has provided. Victor is a visionary supervisor, and his encouragement, patience and support have guided me through my Ph.D. life and this thesis. Every discussion with him makes me a better researcher. It is my fortune to have him as a mentor and role model in research and in life. Stephanie Seneff has also been an invaluable mentor in helping to shape and continuously advance my research. The research in this thesis would not have been possible without her, who has provided mentorship, encouragement, advice, and support on a daily basis. It has been a distinct pleasure to work with Stephanie, whose great passion has inspired me unimaginably over the years.

In addition, I am very thankful to the member of my thesis committee, Regina Barzilay. Discussions with Regina have greatly inspired me during my thesis research. This thesis has improved a great deal based on her feedback.

I am particularly indebted to Alex Gruenstein, Ian McGraw and Yushi Xu for helping me with my thesis projects. Alex helped introduce me various aspects of dialogue system development, as well as many tools and techniques used in the field. The prototype restsaurant recommender system inherited heavily from the CityBrowser system that Alex has implemented with other colleagues in the group. Discussions with Ian about WAMI and AMT were extremely helpful in shaping the system implementation and evaluation in this thesis. And Yushi has been greatly involved in the Mandarin Chinese grammar construction and linguistic parsing aspects.

I would like to thank my collaborators in the speech group at Microsoft Research, including Xiao Li, Alex Acero, Ye-Yi Wang, and Patrick Nguyen, whom I have had the pleasure to work with on the query understanding project. The thesis has significantly benefited from this work. Special thanks go to Harry Shum, Alex Acero, and Xiao Li in Microsoft for their mentorship during my summer internships there.

I am very grateful to the staff of the Spoken Language Systems Group (SLS). Jim Glass has always offered valuable suggestions on my research, whether in group meetings, papers for conferences, and this thesis. Thank you to Scott Cyphers and Lee Hetherington for helping me with various software related issues and innumerable technical challenges.

Meetings with Victor were always smoothly organized thanks to Colleen Russell. And thank you to Marcia Davidson for handling many logistical issues and helping with the data collection tasks.

I've also benefited from my interactions with other members in the SLS group. I would like to thank Paul Hsu, Hung-An Chang, and Yaodong Zhang for their help with many research challenges during my thesis work. Thank you to Sean Liu and Alice Li, who have been a great help in the AMT data collection and the drug domain extention. Also thank you to my labmates who have helped to create an enjoyable research atmosphere, including Najim Dehak, Stephen Shum, Timo Mertens, Jackie Lee, Ann Lee, Ekapol Chuangsuwanich, Ken Schutte, Mitch Peabody, and Ibrahim Badr.

This thesis is based on the research supported by the T-party Project, a joint research program between MIT and Quanta Computer Inc. I was also partially supported by a Microsoft Graduate Women's Scholarship provided by Microsoft.

Finally, thank you to my family and my friends for your unlimited love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The Web has been exploding dramatically over the past decade, especially with user-generated-content (UGC). Social networks and community-contributed sites have become pervasive in people's daily life, such as wikis (e.g., Wikipedia), review sites (e.g., Yelp, TripAdvisor), video/photo sharing platforms (e.g., YouTube, Flicker), social networks (e.g., Facebook) and blogs (e.g., Twitter).

At the same time, there is a rapidly-increasing usage of mobile devices such as smart phones and tablets along with the rapid development of application software (Apps). For example, as of October 2011, there are over 940,000 Apps available on various App Stores[1]. More and more people rely on mobile devices to access the Web, especially for updating social networks and visiting online communities.

Helpful as these mobile applications are, the data available on the Web are growing exponentially and it is impossible for people to digest all the information even with instant Web access. To help users obtain information more efficiently, both the

[1] Approximately 500,000 on Apple App store, 400,000 on Android Market, and 40,000 on Windows Phone Marketplace.

information representation and the interface of content access need to be improved. Text-formed representation is not efficient enough because of the limited screen real estate. And speech-based interaction is inadequate as well, as speech has a limited information space (i.e., it is serial and can't be skipped over). Therefore, there is a great demand for a condensed information representation, i.e., information aggregation or summarization. It would be ideal if one can have a virtual assistant that can summarize the information on the Web in a concise manner, and present it to the user in a natural way.

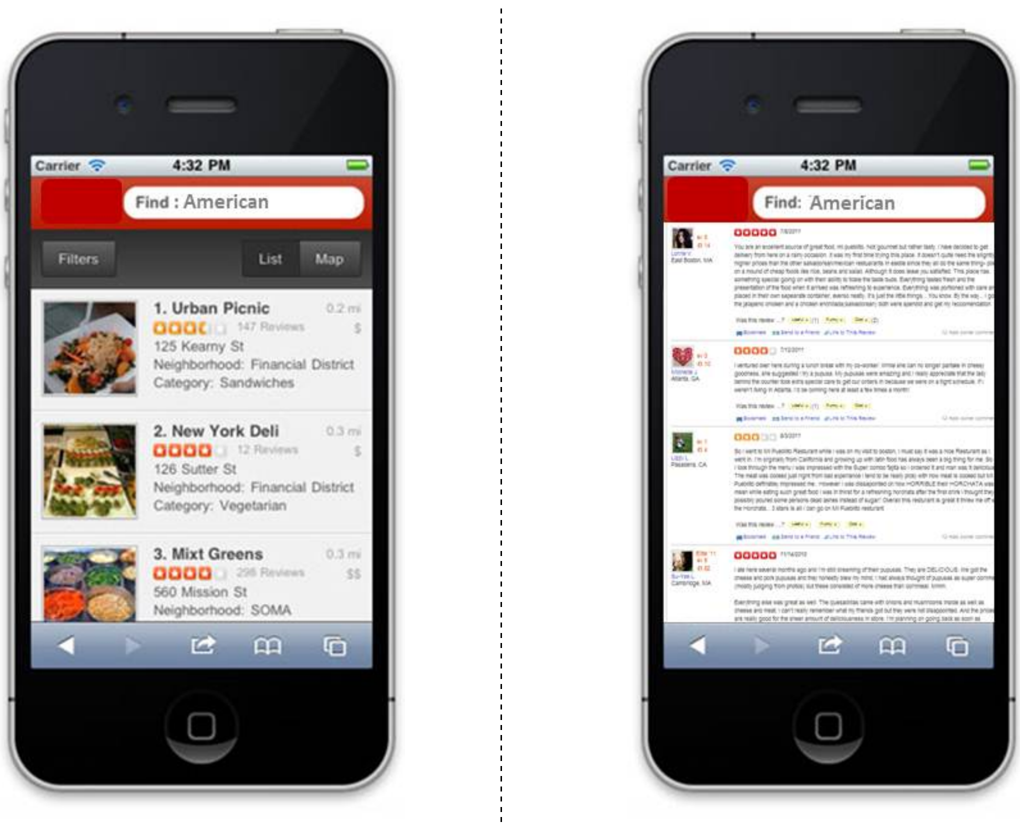

Figure 1-1. Screenshots of a restaurant search App on the mobile phone platform. A few American restaurants are showing up on the screen (on the left-hand side) retrieved by the user's query "American." By clicking on one of these entries, the user could read the reviews on each of these restaurants (on the right-hand side).

The current search paradigm of typing in a search string and obtaining hundreds of relevant hits is still primitive when compared to how humans collaborate to gather information. For example, on the left-hand side of Figure 1-1, the user typed in "American" on a restaurant search iPhone App. Hundreds of American restaurants show up on the screen as the search results, along with hundreds of reviews on each one published by general users (as shown on the right-hand side of Figure 1-1). People could learn the popularity of each restaurant from its rating. But to get more detailed descriptions of these restaurants, they have to read all these reviews. Given that there are hundreds of thousands reviews, it is a rather time-consuming task, not to mention reading them on the small screen.

An intuitive and personalized interface in a more timely informative way could be a human-like assistant system, which can engage a user in a continuous spoken dialogue to learn the user's interest and capture the user's intent proactively, subsequently providing the relevant information in a summarized representation in both spoken and textual forms. For example, Figure 1-2 (the left-hand side) shows an example conversation between a user and a spoken assistant system. The system not only understood the user's questions and found the target restaurants accordingly, but also summarized the 61 entries retrieved in a natural sentence (e.g., "*Many of the rated restaurants have good or excellent ambiance. Most are on Washington Street, Commonwealth Avenue and Market Street*.") The system also summarized the reviews on each restaurant automatically and made recommendations based on the user's interest precisely, for example, finding among these 61 entries the best ones that have "nice ambiance" and "good martinis," as well as giving a high-level description of each of them (e.g., "*Tasca Tapas Restaurant has excellent ambiance, and Wings Express has friendly vibes.*") Figure 1-2 (the right-hand side) also shows the graphical interface of the prototype system (could be on computers, mobile devices or other platforms), locating the target restaurants on a map and showing the condensed review summary on each entry in text (e.g., "*[Food]: wonderful martini, good wine, great fish*" "*[Atmosphere]: romantic date, elegant décor*").

U: Are there any American restaurants in Brighton?

S: There are 61 American restaurants in Brighton. Many of the rated restaurants have good or excellent ambiance. Most are on Washington Street, Commonwealth Avenue and Market Street.

U: Show me the ones that have nice ambiance.

S: I found 4 entries. Devlin's Restaurant and Bar is perfect for romantic date, Stockyard Restaurant has friendly atmosphere, Tasca Tapas Restaurant has excellent ambiance, and Wings Express has friendly vibes.

U: Which one has good martinis?

S: Devlin's Restaurant and Bar serves American food. It has wonderful martinis, good wine, great fish. It is perfect for romantic date. Great place.

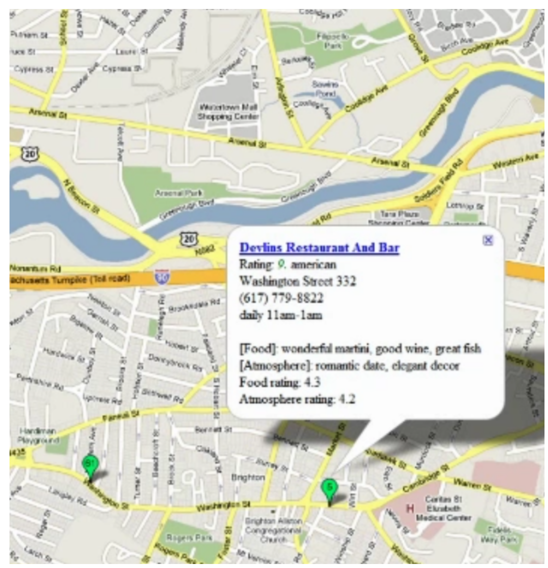

Figure 1-2. An example conversation (on the left-hand side) between a user and a dialogue system, which can aggregate all the reviews on each restaurant and make recommendations to the user based on the summarization over general users' reviews. On the right-hand side is the graphical interface of the prototype system, locating the target restaurant on the map along with highlighted information.

This thesis, therefore, aims to conduct research on a universal framework for developing such conversational systems that can harvest user-generated content and present the summarized information with natural dialogue interaction. The goal is to investigate a platform that marries UGC harvesting and dialogue system development in an effective and portable way. A platform supporting multimodal interfaces for efficient user-generated data access could promisingly benefit people's daily computer interaction experience, as well as potentially advance the technology frontier in industries of consumer electronics and mobile applications.

1.2 Problem Statement

In this section, we will investigate the major challenges in this project. The speech-driven platform can inherit the typical architecture of spoken dialogue systems (as shown in Figure 1-3), consisting of speech-relevant components (speech recognition and speech synthesis), language-relevant components (language understanding and language generation) and dialogue management components (discourse and dialogue modeling). A typical data flow is exemplified with arrows in Figure 1-3. Starting with a user's spoken input, the speech recognizer converts the user's utterance into text and sends it to the natural language understanding (NLU) component, which interprets the user's query and creates a meaning representation. The dialogue manager then searches the database based on the meaning representation and sends the search results to the natural language generation (NLG) component. The speech synthesizer will convert the text-based response created by NLG into speech. The spoken utterance as well as other forms of information (graphs, texts, gestures, etc.) will be sent to the user as the system's response.

Speech processing (recognition and synthesis) have been studied for decades (Rabiner, 1989; Zue and Glass, 2002; Mohria and Pereira, 2002; Gales and Young, 2008), and the techniques have been well developed and widely applied. However, there are still numerous challenges in the *natural language processing* (NLU and NLG), *database harvesting*, and *dialogue modeling* areas (Dialogue and Discourse), especially for supporting sophisticated continuous dialogues. Thus, the focus of this thesis will be on these areas (the right-hand side of Figure 1-3, in the shaded region).

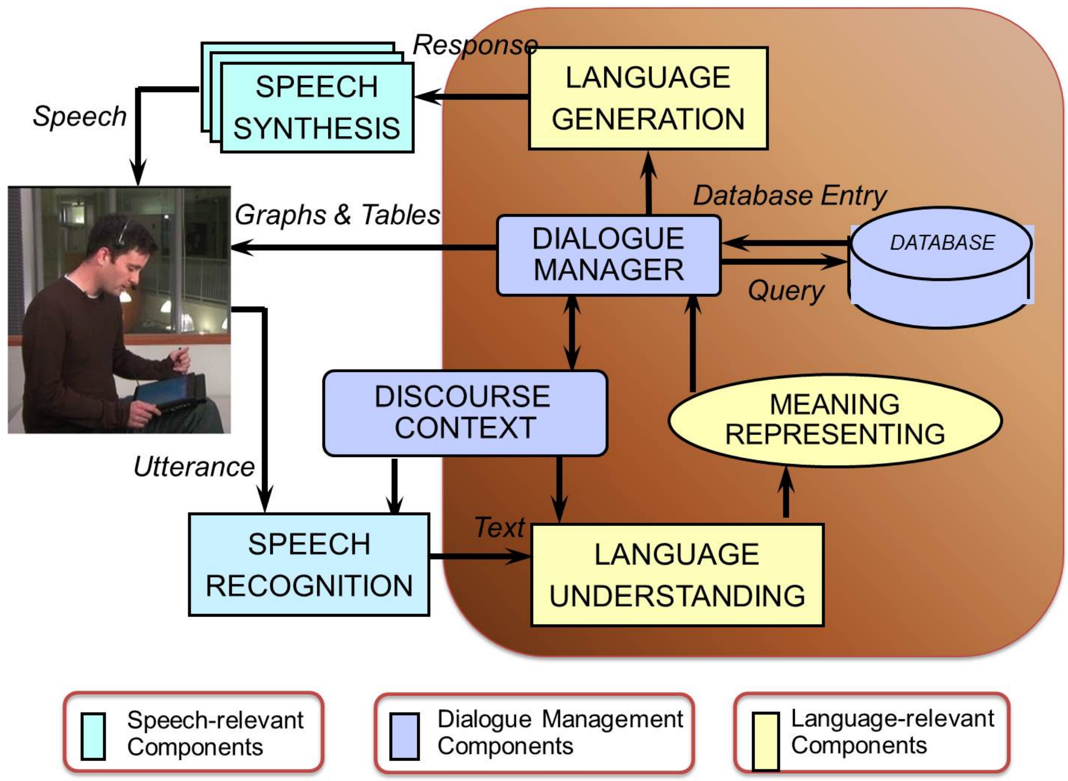

Figure 1-3. Typical architecture of spoken dialogue systems, consisting of speech-relevant components (Speech Recognition and Speech Synthesis), language-relevant components (Language Understanding and Language Generation) and dialogue management components (Discourse and Dialogue Modeling).

A natural user interface, where a user can either type or speak a request, in either keywords or natural language sentences, is a crucial complement of a multimodal spoken dialogue system. Many of the currently existing dialogue systems employ an expert-defined grammar for utterance parsing and language understanding (Seneff, 1992a; Seneff, 1992b; Dowding et al., 1993; Zue et al., 2000; Allen et al., 2005). This could work well in closed-domain systems for fact-answering tasks. However, user-generated data are open-domain data sources and the queries from users for accessing these data can be very diverse and creative. Thus, how to develop a domain-independent and scalable *language understanding* approach to interpret users' queries semantically and to identify the context information as well as capture users' intent correctly is a big challenge.

User-generated content (e.g., public reviews/posts on forums, blogs, and tweets) provide an extensive collection of free-style comments published by general users, which in return provide grassroots-contributions to users interested in a particular topic or service as assistance. But, valuable as they are, user-generated content are unstructured and contain very noisy data, as they were edited by general users freely; not to mention there are hundreds of thousands of community-edited documents available on the Web. Therefore, to filter out context-irrelevant information and to present these unstructured data in a concise dialogue, a *summarization mechanism* is needed to extract the essence from the large number of reviews/posts/tweets and aggregate them into a condensed yet informative summary.

Furthermore, many current dialogue systems are mainly for factoid question-answering tasks, such as weather inquiry (Zue et al., 2000), flight reservation (Seneff and Polifroni, 2000) and bus schedule lookup (Raux et al., 2003). Most of these systems have pre-programmed dialogue templates and are only able to perform restricted dialogue routines in a specific domain. For example, Dell's customer service system (Gorin et al., 1997), "How may I help you," has fourteen pre-defined classes. Upon a user calling, the system classifies each customer call into one of the fourteen classes and responds with a pre-defined dialogue paradigm specifically designed for this class. For more complicated tasks such as aggregated data access, however, the syntax and semantics are very complex, not to mention the ambiguity of discourse in multiple-turn conversation. Thus, we have to go beyond simple question-answering routines and manually-designed response generation templates, and employ a more sophisticated *dialogue modeling mechanism* in order to present the highlighted information of user-generated content in natural and interactive dialogue, as exemplified in Figure 1-2.

Naturally, the task boils down to three problems: 1) how to enhance *language understanding and query interpretation* to capture the users' intent during the conversational interaction; 2) how to equip a standard dialogue system with capabilities of extracting context-relevant information from rich yet *unstructured data* like user-generated content and *summarizing* it into an aggregated form; and 3) how to present the aggregated information to users in *sophisticated dialogues with natural responses*. In this thesis, we will investigate these challenges and try to tackle these problems with an eye on scalable approaches.

1.3 Literature Review

Our work in this thesis therefore draws on a number of research thrusts: natural language understanding for spoken languages, information extraction and aggregation from unstructured data, and dialogue modeling and natural language generation. In this section we will briefly review the state of the art in these areas.

1.3.1 Spoken Language Understanding

Recently, there has been an emergence of personal assistant systems, such as Apple Siri[2]. These systems shed light on the next generation of spoken dialogue systems, which can behave like personal assistants and help users with customized tasks such as hotel booking or restaurant reservation. In such a system, natural language understanding is unified with keyword query understanding for information retrieval and task completion applications. The NLU component of these systems is required to be able to process both spoken and textual queries and identify users' intentions from the extracted meaning representation.

Accurately extracting user intent from typed or spoken queries is a very difficult challenge. Béchet (2008) did a survey on spoken language understanding and classified the task into three major classes: *classification*, *parsing* and *semantic tagging*. As described in Béchet's work, a *classification* task for language understanding is to map a speech message to a label using classifiers, such as Boosting and SVM (Support Vector Machines) in Dell's "How May I Help You" system (Gorin et al., 1997), semantic classification trees on ATIS (Airline Travel Information System) benchmarks (Kuhn and De Mori, 1995), and dialogue act tagging in the CALO meeting system (Tur et al., 2008). A *parsing* task for language understanding is to produce syntactic trees and map them to semantic trees, such as the robust parsing (Seneff, 1992b) and deep semantic understanding (Allen et al., 2005). A *semantic tagging* task is to map a sequence of words to a sequence of attribute/value tokens, using sequential tagging approaches such as

[2] http://www.apple.com/iphone/features/siri.html

Hidden Markov Models (Della Pietra et al., 1998) and Conditional Random Fields (Raymond and Riccardi, 2007; Hahn et al., 2008).

Classification for language understanding is a coarse-grained task mostly on the sentence-level. Recently, more and more studies are focused on fine-grained language understanding, i.e., parsing and semantic tagging on the chunk or slot level. For example, some studies have explored the *combination* of linguistic methods and statistical models. He and Young (2005) proposed a Hidden Vector State (HVS) Model, an extension of the basic discrete Markov model. When used as a semantic parser, the model can capture hierarchical structure without the use of treebank data for training and it can be trained automatically using expectation-maximization (EM) from only-lightly annotated training data. Wang et al. (2009) evaluated several linguistic and statistical techniques to extract user intent from typed sentences in the context of the well-known ATIS domain. They showed that a Semantic Context Free Grammar (CFG) semi-automatically derived from labeled data can offer very good results, and they evaluated several statistical pattern recognition techniques including SVM, Naïve Bayes classifiers and task-dependent *n*-gram language models, which have proved to obtain very low slot error rates if used in combination with the CFG system.

A widely used approach for semantic tagging tasks is *Conditional Random Fields* (*CRF*) *models* (Sarawagi and Cohen, 2004), and there have been a lot of studies recently on how to combine CRF with other statistical methods to improve the performance of language understanding. For example, Raymond and Riccardi (2008) investigated two alternative noise-robust active learning strategies that are either data-intensive or supervision-intensive. They applied uncertainty based active learning with CRF on the concept segmentation task for spoken language understanding and performed annotation experiments on two databases, ATIS (English) and MEDIA (French). Dinarelli et al. (2009) proposed discriminative re-ranking of concept annotation to jointly exploit generative and discriminative models. They improved the FST (Finite State Transducer)-based generative approach, which is a state-of-the-art model for the LUNA corpus (Dinarelli et al., 2009). The re-ranking model also improves FST and CRF on the MEDIA corpus (Bonneau-Maynard et al., 2005) when small data sets are used. Recently, Li et al. (2009) investigated both supervised and semi-supervised learning approaches to web query tagging using Markov and semi-Markov CRFs. In particular, Li (2010) showed that combinations of lexical features, syntactic features, and lexicon-based semantic features for enhancing tagging performance with semi-Markov CRF based models, appear to be a promising future research direction on spoken language understanding.

In this work, we will extend Li's work (2010) and investigate a new lexicon modeling approach for spoken/textual query tagging. We will explore how to utilize external resources such as query logs for automatic lexicon expansion and weighting in order to enrich the semantic features for semi-Markov CRF models. The approach is domain-independent and can be applied to semantic tagging on both keywords and natural language queries, which is a promising approach towards a better spoken language understanding performance.

1.3.2 Unstructured Data Summarization

Summarization and opinion mining from user-generated content has been well studied for years, with many interesting derived topics. As described in Bing Liu's book "Web Data Mining" (Liu, 2011), the opinion mining task can be classified into three levels: *document-based*, *sentence-based* and *feature-based*. At the *document* level, the common task is to identify whether the sentiment of a user-edited document (e.g., a review/post, a tweet) is positive, negative, or neutral. Both supervised and unsupervised learning methods have been explored for this task (Turney, 2002; Pang et al., 2002; Dave et al., 2003; Mullen and Collier, 2004; Pang and Lee, 2004; Gamon et al., 2005; Pang and Lee, 2005; Chaovalit and Zhou, 2005; Cui et al., 2006; Goldberg and Zhu, 2006; Ng et al., 2006). At the *sentence* level, two sub-level tasks can be derived: subjectivity classification (identifying whether a sentence in a document is subjective or objective), and sentiment classification (identifying whether a subjective sentence is positive, negative or neutral). Various machine learning approaches have been explored for this task as well (Wiebe et al., 1999; Hatzivassiloglou and Wiebe, 2000; Yu and Hazivassiloglou, 2003; Wilson et al., 2005; Wiebe and Riloff, 2005; Kim and Hovy, 2006).

At the *feature* level, the task is finer-grained and more complicated. First, features are identified and extracted from sentences of user-edited documents. Then, the opinions on these features are identified as positive, negative or neutral (polarity identification), and feature synonyms are aggregated. At last, a feature-based opinion summary of multiple documents is further produced. Shallow parsing features such as part-of-speech and pattern matching methods have been used for feature extraction (Morinaga et al., 2002;

Liu et al., 2003; Yi et al., 2003; Hu and Liu, 2004a; Popescu and Etzioni, 2005; Ku et al., 2005; Carenini et al., 2006; Kim and Hovy, 2006; Eguchi and Lavrendo, 2006; Zhuang et al., 2006).

More recently, there have been more studies on multi-facet summarization using topic models and other statistical methods (Goldberg and Zhu, 2006; Snyder and Barzilay, 2007; Higashinaka et al., 2006; Higashinaka et al., 2007; Mei et al., 2007; Titov and McDonald, 2008a; Titov and McDonald, 2008b; Branavan et al., 2008; Baccianella et al., 2009). For example, Titov and McDonald (2008a, 2008b) proposed a joint model of text and aspect ratings that utilizes a modified Latent Dirichlet Allocation (LDA) topic model to build topics that are representative of various aspects, and builds a set of sentiment predictors. Branavan et al. (2008) proposed a method for leveraging unstructured annotations in product reviews to infer semantic document properties, by clustering user annotations into semantic properties and tying the induced clusters to hidden topics in the text. Baccianella et al. (2009) conducted a study on multi-facet rating of product reviews with special emphasis on how to generate vectorial representations of the text by means of POS (Part-Of-Speech) tagging, sentiment analysis, and feature selection for ordinal regression learning. And Sauper et al. (2010) investigated how modeling content structure can benefit text analysis applictions such as extractive summarization and sentiment analysis. They presented a framework to allow the joint learning of an unsupervised latent content model with a supervisded task-specific model.

Summarization techniques, when applied to spoken dialogue systems, however, are much more complicated than those in pure-text systems. In a text-based system, users can browse through multiple reviews and obtain information very quickly by scanning the text. In contrast, when interacting with spoken dialogue systems, the information space (i.e., the number of words) in a dialogue turn is often very limited. As speech is inherently serial and cannot be skipped and scanned easily. The information feedback from the system is only a couple of utterances spoken by the system. A dialogue system which speaks long diatribes in each single conversation turn would likely not be well received. Thus, the generally used review summarization techniques, although very effective in text-based systems, are not quite suitable for interactive dialogue systems. The missing piece is an *interactive dialogue oriented*, *fine-grained*, *informative* yet *condensed* review summarization mechanism.

Therefore, there is a great need for a mechanism that can analyze user-generated content with statistical features as well as linguistic features to create concise summaries for dialogue purposes. In this thesis, we will investigate such a fine-grained feature-level approach to dialogue-oriented unstructured data summarization.

1.3.3 Dialogue Systems and Language Generation

Spoken dialogue systems are presently available both in laboratories and commercially for many purposes. For example, Eckert et al. (1993) developed a dialogue system for train timetable inquiry. Zue et al. (2000) developed a weather inquiry dialogue system ("Jupiter"), which can help users inquire about the weather conditions of many cities in the U.S. Seneff and Polifroni (2000) developed a flight domain dialogue system ("Mercury"), which can help people make flight reservations via interactive dialogue. Raux et al. (2003) developed a dialogue system ("Let's Go"), to allow users to look up bus schedules and seek route guidance. Bohus and Rudnicky (2003) developed

RavenClaw, a plan-based, task-independent dialog management framework. Wahlster (2006) developed SmartWeb, a foundation for multimodal user interfaces to distributed and composable semantic Web services on mobile devices. And Weng et al. (2006) developed a robust, wide-coverage and cognitive load-sensitive spoken dialog interface, "CHAT," which is a conversational helper for automotive tasks.

There are also some groups who have developed interesting multimodal applications, for example, backed by a geographical database. Gustafson et al. (2000) developed a multimodal conversational dialogue system in an apartment domain ("AdApt"). Johnston et al. (2002) developed a multimodal dialogue system, "MATCH" (Multimodal Access To City Help), which provides a mobile multimodal speech-pen interface to restaurant and subway information for New York City. Gruenstein and Seneff (2007) developed a web-based multimodal spoken dialogue system, "CityBrowser," which can provide users with information about various landmarks in major cities in the U.S. via speech, text and gestures. Balchandran et al. (2009) developed a mixed-initiative dialog system for address recognition that lets users specify a complete address in a single sentence with address components spoken in their natural sequence.

Spoken systems for education such as language learning and for assistance to elderly people are also interesting topics. For example, Seneff (2007) developed a dialogue system for Mandarin Chinese learning via multimodal speech-based interaction. Beskow et al. (2009) developed a multimodal spoken dialogue system, the MonAMI Reminder, which can assist elderly and disabled people in organizing and initiating their daily activities. Some groups have also been experimenting with the adaptation of machine learning approaches into dialogue management. For example, Varges et al. (2009)

developed a spoken dialogue system based on reinforcement learning that goes beyond standard rule-based models and computes online decisions of the best dialogue moves. Bridging the gap between handcrafted (e.g., rule-based) and adaptive (e.g., based on Partially Observable Markov Decision Processes - POMDP) dialogue models, this prototype is able to learn high rewarding policies in a number of dialogue situations.

Natural language generation (NLG) has been a major challenge in the development of spoken dialogue systems. The commonly used NLG mechanism is a template-based or rule-based approach. For example, a general-purpose rule-based generation system was developed by Elhadad and Robin (1992). Henschel and Bateman (1997) described a lower cost and efficient generation system for a specific application using an automatically customized sub-grammar. Busemann and Horacek (1998) proposed a system that mixes templates and rule-based generation, and Stent (1999) proposed a similar approach for a spoken dialogue system.

Rule-based and template-based generation methods require a lot of human effort and expert knowledge. A more scalable approach for language generation is corpus-based methods, which have also been employed in dialogue systems. For example, Oh and Rudnicky (2000) proposed a corpus-based NLG system, where they model language spoken by domain experts performing the task of interest, and use that model to stochastically generate system utterances. They have applied this technique to sentence realization and content planning, and have incorporated the resulting generation component into a working natural dialogue system. Also, Rambow et al. (2001) showed how the high cost of hand-crafting knowledge-based generation systems can be overcome by employing machine learning techniques. In their framework, NLG was conceptualized as a process leading from a high-level communicative goal to a sequence of communicative acts which accomplish this communicative goal.

In this work, we will discuss a new NLG approach combining template-based and corpus-based methods, which could learn linguistic patterns of sentences automatically from an external corpus and create predicate-topic relations probabilistically, therefore bringing in much flexibility to natural language generation.

1.4 Framework Overview

In the rest of this thesis, we will investigate the problems existing in the state of the art of these areas we have discussed, and propose some approaches to tackle these challenges. In Chapter 2, we will describe a lexicon modeling approach for spoken language understanding. We will explore how to utilize external resources such as search queries from general users to better understand users' spoken/textual input. In Chapter 3, we will explain the unstructured data aggregation process, with a combination of linguistic and statistical approaches to analyzing the semantic and the sentiment of data as well as generating a summarized database. Figure 1-4 (the bottom layer) shows the pipeline of the process. Briefly speaking, user-generated documents will be subjected to a linguistic parser for context-relevant phrase extraction, and the sentiment degree of the extracted expressions can be assessed by a cumulative offset model. A classification model can be used to select high-quality phrases for further topic clustering and aspect rating, in order to create a summary database that can be accessed by the dialogue system (the upper layer of Figure 1-4).

In Chapter 4, we will explain our efforts on developing a dialogue modeling mechanism to support sentiment-involved recommendation-like conversations, as well as a corpus-based predicate-topic selection method for automatic natural language generation. Chapter 5 will explain the experiments on the proposed approaches with real data and describe the implementation of a prototype system. In Chapter 6, we will discuss the portability of the framework and explore an extended application in a novel domain as well as in another language. Chapter 7 will conclude the thesis and discuss the further work.

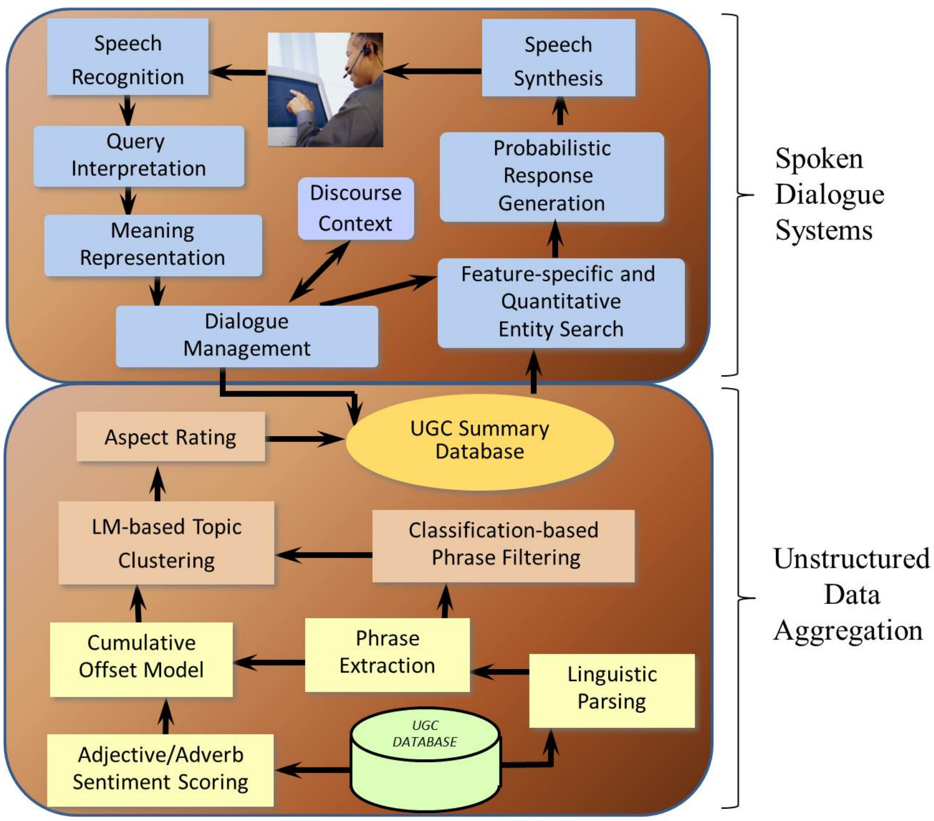

Figure 1-4. The framework of the proposed approaches. The bottom layer is the aggregation process of user-generated content. The upper level is spoken dialogue systems, which look up the UGC summary database for dialogue management and responses generation.

Chapter 2

Language Understanding

For advanced human-computer interaction, natural language understanding is a key component, which is required to be able to process spoken/textual queries and identify users' intentions from users' utterances. For example, given a user's query for requesting movie reviews from online forums (e.g., "show me avatar reviews"), the system should be able to understand that the user's intent is to find reviews of the movie titled "Avatar" and take relevant actions accordingly.

There are three key components required by such a query understanding engine: (1) domain classification; (2) domain-dependent intent detection; and (3) semantic tagging (or slot filling). Figure 2-1 shows an example of language understanding. For example, the query "book me a double room for 2 at Marriott Seattle on Friday" should be classified into the "*Hotel*" domain with the intent "*Make a reservation*." Furthermore, the sentence should be segmented into slots and the semantic of each segment should be identified, e.g., "book me a <*RoomType*>double</*RoomType*> room for <*GuestNumber*>2</*GuestNumber*> at <*HotelName*>Marriott</*HotelName*> in <*Location*>Seattle</*Location*> on <*ReservationDate*>Friday</*ReservationDate*>." Semantic classification at the utterance level (components 1 and 2) has been well studied for years. Segment or slot level semantic tagging (the last component), i.e., segmenting a natural language or keyword query into slots and classifying the slots into semantic roles, however, is a much more complicated task.

Input query
"Book me a double room for 2 at Marriott Seattle on Friday"

Domain	*Hotel*	
User intent	*Make a reservation*	
Semantic tagging	*<RoomType>*	*Double*
	<GuestNumber>	*2*
	<HotelName>	*Marriott*
	<Location>	*Seattle*
	<ReservationDate>	*Friday*

Figure 2-1. Example of natural language query understanding. The input is a query sentence from a user. The output from a language understanding system consists of three parts: "*Domain*," "*User intent*" and "*Semantic tagging*".

There has been a large body of work on semantic tagging in the area of spoken language understanding (Wang et al., 2002; Wang et al., 2005; Pasca and Van Durme, 2007; Hakkani-Tr and Tur, 2007; Hahn et al., 2008; Pasca and Van Durme, 2008; Dinarelli et al., 2009), but relatively few studies are oncerned with keyword query understanding. Only recently, Li et al. (2009) investigated both supervised and semi-supervised learning approaches to web query tagging using Markov and semi-Markov Conditional Random Fields (CRFs) (Sarawagi and Cohen, 2004). In particular, Li (2010) proposed the use of lexical, semantic and syntactic features in semi-Markov CRF-based models, and showed that lexicon-based semantic features are crucial in enhancing semantic tagging performance.

A lexicon is a dictionary of entities of a certain class, e.g., a list of movie titles or restaurant/hotel names. Table 2-1 gives some examples. Each element in a lexicon is a surface form of an entity. Lexicons are normally collected from structured entity databases. However, such lexicons have limited coverage when used for query tagging, as the surface forms of entities in real users' queries are often different from their formal forms in the structured database. For example, the movie entitled "*the devil wears prada*" is often referred to as "*devil wearing prada*" or "*the devil in prada*" in real users' queries (spoken or written). Furthermore, all the elements in a lexicon are often treated equally when used as semantic features, despite the fact that some surface forms are more popular or ambiguous than others. For example, "*McDonald's*" and "*Neighbors*" both appear in the restaurant name lexicon in Table 2-1; however, in most contexts "*neighbors*" is less likely to be used as a restaurant name than "*McDonald's*".

Restaurant name lexicon	**Movie title lexicon**	**Hotel name lexicon**
McDonald's	*The devil wears prada*	*Best western*
Neighbors	*Peter & the Wolf*	*Super 8*
Sea star restaurant	*Little Women*	*Comfort inn*
Red lobster	*Beauty and the Beast*	*Days inn*
Bamboo garden	*Wonder Boys*	*Hampton inn*
Black angus	*Lost Horizon*	*Motel 6*
Blue ginger restaurant	*American Beauty*	*Americas best value inn*
Brasa restsaurant	*The Awful Truth*	*Budget inn*
Fish cafe	*Easter Parade*	*Residence inn*
Cafe vivace	*Champion*	*Marriott hotels resorts suites*
Chen's chef	*Men in Black*	*Holiday inn express hotel suites*
Crab pot seafood restaurant	*Three Little Pigs*	*Homestead studio suites*
Daniel's broiler	*Design for Death*	*Red carpet inn*
Dave's last resort	*Inglourious Basterds*	*Hyatt hotels resorts*
King buffet	*Heaven Can Wait*	*Embassy suites hotels*
Starbucks	*Father and Daughter*	*Clarion hotel conference center*

Table 2-1. Examples of lexicons: partial lists of restaurant names, movie titles, and hotel names. These lexicons were collected from structured entity databases.

To address these major problems of lexicons: ambiguity, limited coverage and lack of relative importance, we propose a new lexicon modeling method to improve query semantic tagging. In this chapter, we will first explain the task of semantic tagging, and then describe the proposed lexicon modeling approach, including both generative models and discriminative models, for automatic lexicon expansion and weighting using external resources such as web query logs (Liu et al., 2011a).

2.1 Semantic Tagging[3]

Semantic tagging is a widely-used approach to textual/spoken query understanding. The task is to segment a natural language or keyword query into slots and classify the slots into semantic roles (exemplified in Figure 2-1), which is often formulated as a joint segmentation and classification problem (Li, 2010), i.e.,

$$s^* = argmax_s \, p(s|x) \tag{2.1}$$

where $x = (x_1, x_2, \dots, x_M)$ is an input word sequence. The goal is to find $s = (s_1, s_2, \dots, s_N)$, which denotes a segmentation of the input as well as a classification of all segments. Each segment is represented by a tuple $s_j = (u_j, v_j, y_j)$. Here u_j and v_j are the start and end indices of the segment, and y_j is a class label. The segment sequence can be augmented with two special tokens, *Start* and *End*, represented by s_0 and s_{N+1}, respectively.

[3] This work was done at Microsoft Research in collaboration with Xiao Li, Alex Acero and Ye-Yi Wang.

İn previous studies, CRF models were often used as the segmentation/classification model for sematic tagging (Raymond and Riccardi, 2008; Dinarelli et al., 2009). Particularly, Li et al. (2009) investigated both supervised and semi-supervised learning approaches to semantic tagging using semi-Markov CRF models:

$$p(s|x) = \frac{1}{Z_\lambda(x)} exp\left\{\sum_{j=1}^{N+1} \lambda \cdot f(s_{j-1},\ s_j,\ x)\right\} \tag{2.2}$$

where the partition function $Z_\lambda(x)$ is a normalization factor; λ is a weight vector; and $f(s_{j-1},\ s_j,\ x)$ is a vector of feature functions defined on segments. More precisely, f is of the functional form $f(y_{j-1}, y_j, x, u_j, v_j)$. Given manually-labeled queries, the goal is to estimate λ that maximizes the conditional likelihood of training data while regularizing model parameters. The learned model is then used to predict the label sequence s for future input sequence x.

Li (2010) also investigated the use of transition features, lexical features, semantic features and syntactic features in semi-Markov CRFs, and showed that semantic features are very critical, especially for named entity recognition. An effective way of constructing semantic features is to inspect whether a hypothesized query segment matches any element in a given lexicon. In other words, the feature value is given by:

$$f(s_{j-1},\ s_j,\ x) = \delta(s_j \epsilon\ L)\delta(y_j = b) \tag{2.3}$$

where L denotes a lexicon, b denotes a class, and $\delta(s_j \epsilon\ L)$ denotes that the current segment matches an element in lexicon L, which we refer to as "exact match".

The semantic features using exact match of lexicons were proved to be very useful. However, they have some limitations. First of all, the surface form of entities in users' queries may not be the same as in the lexicon, which was collected from a formal database. For example, a lexicon pre-collected from a restaurant database contains an element "*Bamboo Garden Chinese restaurant*," while in users' queries it is often referred to as "*Bamboo Garden restaurant*" or "*Bamboo Garden*," which cannot be discovered using exact match. Also, lexicons collected from pre-existing databases are often incomplete. The databases might not be regularly updated and there might be many new entities that are not included in the databases (e.g., new-released movies, new-opened restaurants). Furthermore, among all the elements of a lexicon, some are often more popular or ambiguous than others; thus, the elements in a lexicon should not be treated equally when used as semantic features.

To tackle these problems, we will explore how to better utilize pre-collected lexicons and automatically expand lexicons to overcome the short coverage of pre-existing databases, as well as learning different weights of lexicon elements from external resources such as web query logs.

2.2 Lexicon Modeling

A natural way of increasing the lexicon coverage is to use *fuzzy match* features instead of exact match features (as described in Equation (2.3)). To extract fuzzy match features, we take the maximum "similarity" between a query segment and all lexicon elements as the feature value, instead of a binary feature. Specifically, we treat each lexicon element as a "document" and compute the idf (inverse document frequency) (Salton and Buckley,

1988) score of each word type accordingly. Let v_{s_j} and v_l denote the tf-idf (term frequency–inverse document frequency) (Salton and Buckley, 1988) vector of a query segment and that of a lexicon element, respectively. The fuzzy match feature value is computed as:

$$f(s_{j-1},\ s_j,\ x) = \max_{l \in L} \frac{v_{s_j} \cdot v_l}{|v_{s_j}||v_l|} \cdot \delta(y_j = b) \tag{2.4}$$

where L denotes a lexicon, b denotes a class, and $\frac{v_{s_j} \cdot v_l}{|v_{s_j}||v_l|}$ denotes the cosine distance between the tf-idf vector of current segment and that of an element in lexicon L.

Such fuzzy match features $f(s_{j-1},\ s_j,\ x)$ will be used for semi-CRF model training (as in Equation (2.2)). Fuzzy match is easy to implement as no change is required to the original lexicons. But the computation cost might be high. Exact match is just a one-step table lookup. For fuzzy match, however, a segment has to be compared with each element in the lexicon that has any words overlapping with the segment, for tf-idf similarity computation (as visualized in Figure 2-2). Given a segment containing a word that is very common in a lexicon, the computation can be expensive as an online operation.

A more efficient way of employing semantic features is to obtain an expanded lexicon that has a higher coverage. To automatically learn new lexicons, we leverage external web resources such as query logs which contain a very large collection of user-submitted queries that cover various surface forms of a large amount of entities. İn the following sub-sections, we will explore two types of models, generative models and discriminative models, for automatic lexicon expansion and weighting.

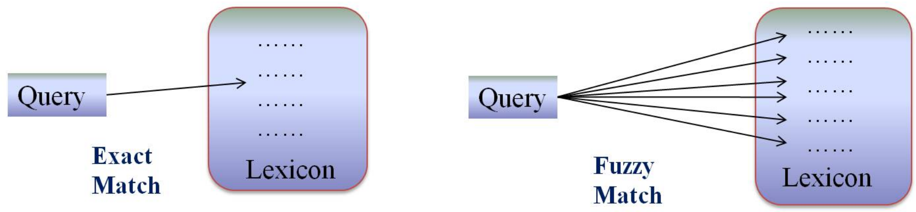

Figure 2-2. Visualization of exact match and fuzzy match on a query segment given a lexicon. On the left-hand side is the one-to-one extract match, and on the right-hand side is the one-to-many fuzzy match.

2.2.1 Generative Models

First, we will explore how to learn new lexicns from web query logs. Here, we let y denote the entity class such as "*HotelName*" and "*MovieTitle*," w denote the surface form of an entity, and d_y denote an external resource (e.g., a web document) that corresponds to an entity in the entity class y. Given a class y and a pre-collected lexicon belonging to this class, the goal is to expand the original lexicon by learning new entities or new surface forms of entities w through d_y.

Table 2-2 shows an example of partial query logs from a search engine. Each row is an event of query submission and web document click by users. "#Click" represents the number of click events among all the users during the period over which the query logs were collected. There are three major *characteristics* of user query logs that make them a perfect resource for lexicon learning and weighting. First, the web documents that users clicked are normally *relevant* to the queries they submitted. Second, the statistics of user clicks vary among different *queries*, which shows that different entities may have different popularities (e.g., a more popular movie might be queried more often and thus has more clicks). Third, statistics of user clicks vary among different *surface forms* of a query, which indicates that different surface forms of an entity may have different popularities as well (e.g., Query 1 and Query 4 in Table 2-2 both inquire about the same movie and both have the same web document clicked (same title and same URL), but the query "*the devil wears prada*" has many more click events than "*devil and prada*," indicating that the former one is a more popular expression of the movie title among general users).

ID	Query	Title of clicked web document	URL of clicked web document	#Click
1	*the devil wears prada*	*The Devil Wears Prada (2006)*	*http://www.**imdb.com/title**/tt0458352/*	*108*
2	*the devil wears prada quote*	*The Devil Wears Prada (2006) - Quotes*	*http://www.**imdb.com/title**/tt0458352/ quotes*	*9*
3	*the devil wears prada cast*	*The Devil Wears Prada (2006) - Full cast and crew*	*http://akas.**imdb.com/title**/tt0458352/ fullcredits*	*2*
4	*devil and prada*	*The Devil Wears Prada (2006)*	*http://www.imdb.com/title/tt0458352/*	*7*
5	*devil in prada soundtrack*	*The Devil Wears Prada (2006) - Soundtracks*	*http://uk.imdb.com/title/tt0458352/ soundtrack*	*2*
6	*the cay quote*	*The Cay (1974) (TV) - Memorable quotes*	*http://www.imdb.com/title/tt0246477/ quotes*	*3*
7	*the good shepherd cast*	*The Good Shepherd (2006) - Full cast and crew*	*http://www.imdb.com/title/tt0343737/ fullcredits*	*11*

Table 2-2. An example of a partial query log of a search engine in the movie domain. Each row represents a query submitted by users and the "Title" and "URL" of the web page that users clicked among those retrieved by the search engine. "#Clicks" represents the number of total click events among all the users. Rows above the double bars are extracted from a pre-existing lexicon as query seeds, and the rows below the double bars are newly discovered queries.

Given these characteristics, a plausible approach is to learn new lexicon elements from user-submitted queries based on the relevance between queries and web documents, as well as to learn different weights for each lexicon element from the statistics of users' clicks. Specifically, to identify the set of web documents d_y relevant to an entity class y, we use pre-collected lexicon elements as query seeds and find web documents relevant to these queries from query logs. We then extract common patterns from these documents, which can be regular expressions in URLs or keywords in the titles of documents. Next, we identify new documents d_y from query logs that match the learned patterns as potential documents relevant to the domain. Then from queries relevant to these newly-discovered documents, we could extract unseen surface forms w as new lexicon candidates.

Take the movie domain as an example. Our task is to expand the "*MovieTitle*" lexicon which ideally would contain all surface forms of all movie titles. Assume we have a pre-collected lexicon available, which contains an element "*the devil wears prada.*" We will use the elements in this pre-existing lexicon as seeds and filter the query log for queries that contain these seeds. As shown in Table 2-2, the queries above the double bars (Query 1, 2 and 3) represent the set of queries that match the lexicon seed. Among the documents that were clicked by users after issuing these queries, a common pattern in the URLs is "*imdb.com/title.*" This is reasonable as the query seeds are from a lexicon of "*MovieTitle*," and "IMDB" is a popular web site for movies. Thus, we assume that URLs sharing the same pattern are most likely relevant to the domain, and we match this pattern with all the URLs in the query log, extracting those rooted at "*imdb.com/title*" as well (in Table 2-2, the newly-discovered URLs are those below the double bars). These newly-discovered web documents are considered as relevant to the lexicon class ("*MovieTitle*"). Then, from the query logs we extract all the rows which contain these documents, and the query in each row (Query 4, 5, 6 and 7) is considered as a new lexicon candicate relevant to the class.

Patterns can also be learned from document titles. For example, we desire to expand a lexicon in the restaurant domain that contains restaurant names. Most clicked documents corresponding to queries that contain restaurant names might contain domain-relevant contextual keywords in their titles. We still take the pre-collected lexicon as query seeds. For example, the queries above the double bars in Table 2-3 contain pre-existing lexicon elements "*olive garden*," "*hyde park*" and "*silver fox*." We extract the documents that are relevant to these queries from the query logs (Row 1, 2 and 3), and learn the most frequent contextual keywords (e.g., "*restaurant*," "*steakhouse*" and "*bar & grill*") from these relevant dcuments. Then, we filter the query logs to discover new documents whose titles contain these keywords as well (e.g., the web documents below the double bars in Table 2-3). These documents are considered as domain-relevant. We then extract new lexicon candidates from queries (e.g., Query 4 and 5) that are relevant to these newly-discovered documents.

The new lexicon candidates learned from users' queries often co-occur with some context words (e.g., "*quotes*" and "*cast*" for movies in Table 2-2; "*coupons*" and "*menu*" for restaurants in Table 2-3). To get clean surface forms that do not contain such context words, we use pre-existing lexicon elements as seeds and extract from query logs the set of queries that contain these seeds. We then replace the lexicon seeds appearing in these queries into a uniform expression (e.g., "*<movie title> quotes*"; "*<movie title> cast*") and learn the most popular query context words based on frequency statistics (e.g., "*quotes*" and "*cast*"). And these context words will be removed from the learned lexicon candidates.

ID	Query	Title of clicked web page	URL of clicked web page	#Clicks
1	*olive garden coupon*	*Olive Garden Italian Restaurant - Zip Locator*	*http://www.olivegarden.com/locate.asp*	*8*
2	*hyde park menu*	*Hyde Park Bar & Grill - since 1982 - Menu*	*http://www.hydeparkbarandgrill.com/menu.html*	*3*
3	*silver fox*	*Silver Fox Steakhouse | Richardson Seafood Restaurant*	*http://www.silverfoxcafe.com/richardson_location.php*	*2*
4	*vincents nyc coupon*	*Vincent's - New York Restaurant - MenuPages Italian, Pizza Restaurant Search*	*http://www.menupages.com/restaurants/vincents/*	*2*
5	*shuckers restaurant menu*	*Shuckers Oyster Bar and Grill - Wake Forest - Southern Pines*	*http://www.shuckersgrill.com/*	*6*

Table 2-3. Example of partial query log in the restaurant domain. Each row is an event of query submission and web document click by users. "#Click" represents the number of click events among all the users during the period when the query logs were collected. The queries above the double bars contain lexicon seeds from a pre-collected lexicon, and those below are newly learned lexicon candicates.

Given the cleaned surface forms, the final stage is lexicon weighting, i.e., how to estimate the relevance of a surface form to a given lexicon class. In the generative model, we use the probability of a surface form given a class ($p(w|y)$) as the weight. The log probability is calculated by:

$$p(w|y) = \sum_{d_y} p(w|d_y) \cdot p(d_y|y) \tag{2.5}$$

where $p(d_y|y)$ represents the popularity of the document d_y with respect to the entity class, and $p(w|d_y)$ represents the probability of a surface form given a document. The popularity of the document and the probability of the surface form can be learned from user click statistics.

Intuitive examples are shown in Table 2-2 and 2-3. Formally, the probability of a relevant document d_y (e.g., a URL or a title) given a class *y* is defined as the ratio of click count on d_y over the click count on all the documents d_y^* relevant to the class:

$$p(d_y|y) = \frac{click(d_y)}{\sum_{d_y^*} click(d_y^*)} \quad (2.6)$$

And the probability of a surface form w given a relevant document d_y is defined as the ratio of click count on d_y triggered by query w over the total count of clicks on d_y triggered by all the relevant queries:

$$p(w|d_y) = \frac{click(w, d_y)}{click(d_y)} \quad (2.7)$$

With these user click statistics, $p(w|y)$ can be estimated through Equation (2.5), (2.6) and (2.7) for each lexicon element. The normalized probability scores can be used as features for semi-CRF training (as in Equation (2-2)). Together with the expanded lexicon, the weighting features can be used for semantic tagging with semi-CRF models.

A limitation of generative models, however, is that they fail to reflect ambiguity. A lexicon element with high likelihood score may be very confusable with other entity types. For example, “*Paris*” is a popular hotel in Las Vegas, but it is highly confusable with the lexicon of “*CityName*” as it may have high weights in both lexicons. Such a problem can be potentially tackled by a discriminative model.

2.2.2 Discriminative Models

In the discriminative model, we intend to change the feature value while still using the lexicon expanded with the generative model approach. The normalized log posterior probability $p(y|w)$ is used as the feature value:

$$p(y|w) = \sum_{d_w} p(y|d_w) \cdot p(d_w|w) \tag{2.8}$$

where d_w is a web document relevant to the surface form w; $p(d_w|w)$ is a query-document relevance model, representing the probability of a document given a query; and $p(y|d_w)$ is a context document classification model, representing the probability that d_w belongs to class y.

Here, we model the probability of a document given a query ($p(d_w|w)$) as the relevance of the document to the query, which can be learned from query logs. We assume that each query-document pair in query logs is an association pair, i.e., each query and the document that users clicked after issuing this query are relevant. Thus, to learn the document-query relevance from query logs, we use a binary feature $\delta(d_w|w)$ (as an extreme case of $p(d_w|w)$) as a simplification, which is assigned a value 1 if a document d_w co-occured with a query w in the query logs and 0 if not, indicating whether the document is relevant to the query or not.

We model $p(y|d_w)$ as a document classification problem. d_w denotes a document, and y denotes a class label. Specifically, we use the "snippets" of web documents as d_w. Snippet is the short summary of a web document showing up along with each hyperlink retrieved by a search engine. Figure 2-3 shows an example of snippets relevant to a certain query "*bourne identity*" on the Bing search engine.

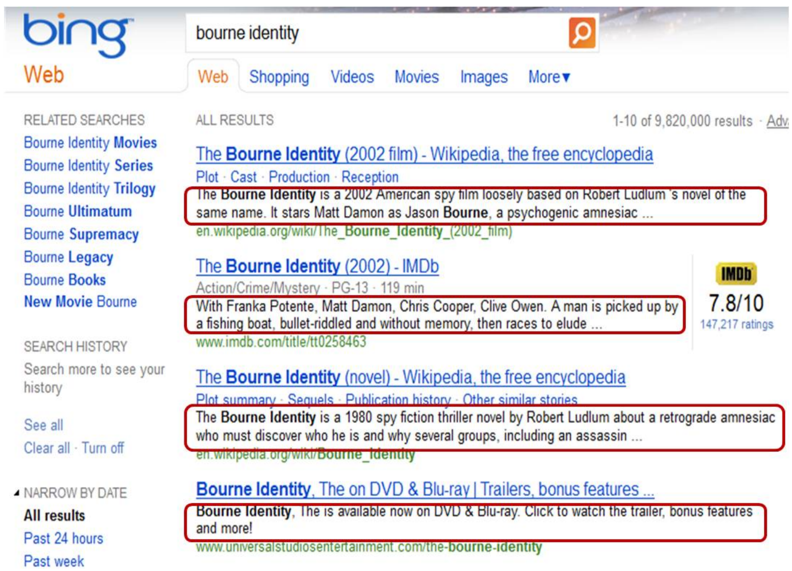

Figure 2-3. An example of snippets of web pages retrieved by the Bing search engine from a query "bourne identity." The blue hyperlinks are the ranked web pages relevant to the query. The short texts under each blue hyperlink (enclosed in red) are the snippets of the retrieved web pages, to show users a brief peek into the summarized content of the web.

Snippets are used instead of the whole web documents for document classification, because a snippet is a condensed summary of each web document, and the density of domain-relevant information (e.g., context-relevant keywords) in snippets is much higher than that in the whole documents. For example, Table 2-4 shows the top-3 ranked snippets relevant to two queries, "*bourne identity*" and "*dances with wolves*," retrieved from the Bing search engine. These lists can be harvested from web query logs, which provide the correlation between queries and snippets as well as the ranking positions of snippets by the relevance to each query. These snippets are summaries of the highest-ranked web documents for each query, and therefore contain rich context-relevant information (e.g., context keywords like "*film*," "*stars*," "*thriller*," "*epic*" and "*Oscars*" occur frequently in the example snippets as the queries are elements of the "*MovieTitle*" lexicon).

Ranking	**Query**	**Snippet of retrieved web page**
1	*bourne identity*	*The Bourne Identity is a 2002 American spy film loosely based on Robert Ludlum's novel of the same name. It stars Matt Damon as Jason Bourne, a psychogenic amnesiac…*
2	*bourne identity*	*With Franka Potente, Matt Damon, Chris Cooper, Clive Owen. A man is picked up by a fishing boat, bullet-riddled and without memory, then races to elude ...*
3	*bourne identity*	*The Bourne Identity is a 1980 spy fiction thriller novel by Robert Ludlum about a retrograde amnesiac who must discover who he is and why several groups, including an assassin ...*
1	*dances with wolves*	*Dances with Wolves is a 1990 epic film based on the book of the same name which tells the story of a Civil War era United States Army lieutenant who travels to the American frontier*
2	*dances with wolves*	*Kevin Costner's 1990 epic won a bundle of Oscars for a moving, engrossing story of a white soldier (Costner) who singlehandedly mans a post in the 1870 Dakotas, and ...*
3	*dances with wolves*	*"Dance with the Wolves" is one of Ukrainian singer Ruslana's singles, released in 2005. Two music videos were made for the song. During the shootings, Ruslana had to be in a ..*

Table 2-4. Example of query-snippet pairs extracted from query logs. The column of "Query" represents the queries that users have submitted to the search engine. The columns of "Snippet of clicked web page" represents the short content summary of the Web page that users clicked, and the column of "Ranking" represents the ranking position of each document among the documents retrieved for each query.

Given the expanded lexicon, we view each new lexicon candidate w as a query and take the snippets of top-n ranked documents retrieved on this query from a search engine as d_w. To estimate $p(y|d_w)$, we use the snippets as the "bag-of-words" and train a classification model over lexicon classes. To collect training data for the classficiation model, we use the pre-collected lexicon to generate query seeds, and select the top-n snippets of documents retrieved by each query seed as positive samples (documents that are relevant to the pre-existing lexicon have high correlation with the class). Negative examples are selected by using a set of domain-irrelevant lexicons (e.g., lexicons in other classes such as cities and states) as queries and retrieving top-n snippets on each query as documents that have low relevance to the class. A document classifier can be trained with these samples, using n-grams in the snippets as features.

For each newly-discovered lexicon candidate, we take it as a query and extract from query logs the snippets of the top-n ranked documents relevant to this query. These snippets can be treated as the "bag-of-words" for this query and the learned classifier can be applied for classification. The posterior classification score $p(y|d_w)$ is used as the weight of each lexicon candidate and the normalized score can be employed as semantic features for semi-CRF training (as in Equation (2-2)).

2.3 Chapter Summary

In this chapter, we explored a lexicon modeling approach to automatically learn new lexicon and assign weights to its elements utilizing web query logs. For lexicon expansion, we use a generative model to extract patterns from query logs using known lexicon seeds, and discover new lexicon elements using the learned patterns. For lexicon weighting, we proposed two approaches based on generative and discriminative models to learn the relative importance of lexicon elements from user click statistics. The normalized log probability $(p(w|y))$ learned from generative models and the posterior $(p(y|w))$ from discriminative models are used as the new features for semi-Markov CRF model training.

In Chapter 5 (Section 5.1), we will apply the lexicon modeling approach to a semantic tagging system. We will evaluate the proposed approach with real data in various domains. The experiments will show that these generative and discriminative models for lexicon learning and weighting can significantly improve the performance of semantic tagging on both keywords and natural language queries. The approach is domain-independent and can be flexibly applied to spoken language understanding, therefore helping dialogue systems understand users' intentions robustly.

Chapter 3

Unstructured Data Processing

A speech-navigated information aggregation system should be able to obtain and summarize user-generated content (UGC) from the Web and utilize it as a knowledge base for multimodal data access services. İn this chapter, we will explore approaches to processing unstructured data and inferring from them succinct summaries. An example of user-generated content is shown in Figure 3-1, with two user-published restaurant reviews on “www.citysearch.com”.

Eclectic but awesome *by Alice* ***Rating: 4.5***

- ***Pros:*** *Fantastic food; super friendly staff*
- ***Cons:*** *none really*

This food was fantastic. I didn't go here with stellar expectations, I think b/c I couldn't quite make sense of the menu. I came here for my friend's bday. 5 of us, 2 vegetarians. We ordered the Asian salad…

An Underated Jewel of a Restaurant *by Bob* ***Rating: 5***

- ***Pros:*** *sexy ambience, sassy crowd + satisfying small plates*
- ***Cons:*** *the wait is often too long later in the week.*

By now, most of you know (or otherwise should) that Cuchi Cuchi remains the best place to go when you want both delicious, authentic Spanish Tapas and a warm, romantic setting…

Figure 3-1. User-generated reviews on a restaurant called “Cuchi Cuchi” published on www.citysearch.com. Each review mainly contains a “title,” an “overall rating,” “Pros,” “Cons” and a free-style comment. The real names of the reviewers were replaced by “Alice” and “Bob”.

The goal is to harvest these user-generated data from the Web, and summarize them into a condensed information representation which can be accessed by speech-based applications such as dialogue systems. A possible representation format of information is shown in Figure 3-2, which summarizes the reviews in Figure 3-1 in representative aspects (e.g., "*food*," "*service*," "*atmosphere*" and "*general*") as well as calculating an average rating for each aspect.

Aspect	Extracted phrases	Rating
Atmosphere	*sexy ambience, sassy crowd, warm romantic setting*	*4.8*
Food	*satisfying small plates, fantastic food, authentic Spanish Tapas*	*4.1*
Service	*super friendly staff*	*4.3*
General	*awesome restaurant, best place to go, long wait*	*3.8*

Table 3-1. Example of a summary generated from the reviews in Figure 3-1, including representative phrases selected on each aspect and an average rating based on the expressions within each aspect.

To achieve this goal, there are a few problems to tackle. Firstly, the representative phrases (e.g., opinion-related expressions) have to be identified and extracted from the original unstructured data. In this section, we will explore a parse-and-paraphrase paradigm, which utilizes a lexicalized probabilistic syntactic grammar based on well-formed linguistic structure, to identify semantically context-related phrases from user-generated texts.

Secondly, we need to estimate the sentiment in these extracted opinion-related phrases, ideally on a numerical scale, in order to calculate aspect ratings. "Pros" and "Cons" edited by users often provide information about binary polarities (positive and negative). However, human language contains rich vocabularies to express different degrees of sentiment (e.g., "*excellent*," "*great*," "*fair*" and "*not bad*" express different positive levels). İn addition to descriptive adjectives, adverbials also play an important role in determining the degree of the orientation (e.g., "*absolutely*," "*fairly*" and "*a little*" each expresses a different confidence level on the words it modifies). Thus, in this section we will explore how to assess the degree or strength of sentiment in various expressions by modeling both adverbials and adjectives.

Furthermore, to generate a condensed summary of the original unstructured data set, we have to filter out irrelevant or low quality phrases and catalogue the high-quality and relevant phrases into representative aspects. Thus, a third direction we will explore is quality-based phrase classification as well as topic clustering and aspect rating.

3.1 Parse-and-Paraphrase Paradigm for Phrase Extraction

Firstly, we will investigate an approach to extracting opinion-relevant phrases from user-generated content. Some previous work on phrase extraction from user-generated content focused on statistical features, such as extracting adjacent adjective and noun based on POS-tagging and frequency counts (Hu and Liu, 2004b); while linguistic structure such as implicit long-distance dependency is often disregarded (e.g., "The *seafood platter* which people said very good is just *above average*"). High level linguistic features, if well utilized and accurately extracted, can provide much insight into the semantic meaning of user opinions and contribute to sentiment identification.

Another challenging aspect is the proper scoping of negations over the right constituent. The simple way of handling negations is keyword matching: if a negation word such as "not" appears in a sentence, the polarity of sentiment is reversed. However, not in all cases negations represent opposite sentiment (e.g., "This restaurant does *not* only provide great food but also has very nice vibes"), which we argue can be handled well with careful linguistic analysis by examining the semantic structures.

There have been many studies on utilizing topic models for text analysis, especially for user-generated reviews. Latent topics and underlying semantic concepts can be revealed by these methods. For the application of dialogue systems, however, the focus is not only learning the general concepts, but also extracting individual topics from each user-generated document (e.g., "chicken tikka masala," "spaghetti carbonara") in order to provide users with accurate response upon various queries. With linguistic analysis, we could capture such topics even with very low frequency over the entire corpus, which on the other hand are likely to be overlooked by statistical methods.

Driven by these challenges, we propose a linguistic parsing method to extract adverb-adjective-noun phrases based on clause structure obtained by parsing sentences into a hierarchical representation (Liu and Seneff, 2009). Our linguistic analysis is based on a parse-and-paraphrase paradigm. Instead of the flat structure of a surface string, the parser provides a hierarchical representation, which we call a linguistic frame. It preserves linguistic structure by encoding different layers of semantic dependencies. The grammar captures syntactic structure through a set of carefully constructed context free grammar rules, and employs a feature-passing mechanism to enforce long distance constraints. The grammar is lexicalized, and uses a statistical model to rank order competing hypotheses.

The grammar probability model was trained automatically on the corpus of user-generated content.

An example linguistic frame is shown in Figure 3-2, which encodes the sentence "*The caesar with salmon or chicken is really quite good*." In this example, for the adjective "*good*," the nearby noun "*chicken*" would be associated with it if only proximity is considered. From the linguistic frame, however, we can easily associate "*caesar*" with "*good*" by extracting the head of the topic sub-frame and the head of the predicate sub-frame, which are encoded in the same layer (root layer) of the linguistic frame. In this way, long-distance dependencies are taken into consideration based on the semantic structure of sentences.

```
{c cstatement
 :topic {q caesar
        :quantifier "def"
        :pred {p with
                 :topic {q salmon
                        :pred {p conjunction
                                 :or {q chicken }}}}
 :adv "really"
 :pred {p adj_complement
        :pred {p adjective
              :adv "quite"
              :pred {p quality
                       :topic "good" }}}}
```

Figure 3-2. An example of the hierarchical linguistic frame generated for the sentence "*The caesar with salmon or chicken is really quite good*." The topic frame ("*{q caesar*") and the predicate frame ("*{p adj_complement*"}) are on the same level, which indicates the head of the noun phrase should be associated with the adjective complement.

To produce the opinion-relevant phrases, a set of generation rules is carefully constructed to only extract sets of related adverbs, adjectives and nouns. For example, the adjective-noun relationships for opinion-relevant phrases can be captured from the following linguistic patterns: (1) all adjectives attached directly to a noun in a noun phrase, (2) adjectives embedded in a relative clause modifying a noun, and (3) adjectives related to nouns in a subject-predicate relationship in a clause. These patterns are compatible, i.e., if a clause contains both a modifying adjective and a predicate adjective related to the same noun, two adjective-noun pairs are generated by different patterns. As in, "*The <u>efficient</u> <u>waitress</u> was nonetheless very <u>courteous</u>*." It is a "parse-and-paraphrase-like" paradigm: the paraphrase tries to preserve the original words intact, while reordering them and/or duplicating them into multiple noun phrase units. Since they are based on syntactic structure, the generation rules can also be applied in any other domain involving opinions.

Generation rules can also be constructed to extract adverbials that are associated with descriptive adjectives. Take the frame in Figure 3-2 as an example. There is an adverb "*quite*" modifying the head word "*good*" in the predicate sub-frame. The linguistic frame also encodes an adverb "*really*" in the layer immediately above. A set of well-constructed generation grammar can create customized adverb-adjective-noun phrases such as "*quite good caesar*" or "*really quite good caesar*".

As written by Xuehui Wu (2005): "The scope of negation is a complex linguistic phenomenon. It is easy to perceive but hard to be defined from a syntactic point of view. Misunderstanding or ambiguity may occur when the negative scope is not understood clearly and correctly." Interpreting negation in English is not straightforward, and it is often impossible to do correctly without a deep linguistic analysis. Promisingly, a linguistic approach associating long-distance elements with semantic relations can identify whether a negative reference scopes over a complement clause, a predicate clause or the full sentence, and can handle negations semantically. For example, with the hierarchical linguistic frame, the majority semantic rule for negation can be that it scopes over the remainder of its containing clause.

Figure 3-3 shows the linguistic frame for a sentence that contains a negation word: "*Their menu was a good one that didn't try to do too much.*" If not consider the linguistic structure, the appearance of "*not*" will be treated as a negation. However, simply reversing the sentiment of the sentence to negative polarity is wrong, as the sentence actually expresses positive opinion for the topic "*menu*." But with the hierarchical linguistic frame, the negation "*not*" can be identified as under the sub-frame of the complement clause, instead of on the same or higher layer of the adjective sub-frame; thus it can be considered as unrelated to the adjective "*good*".

```
{c cstatement
:topic {q menu
            :poss "their" } }
  :complement {q pronoun
        :name "one"
        :adj_clause {c cstatement
                       :conjn "that"
                       :negate "not"
                       :pred {p try
                             :to_clause  {p do
                                          :topic {q object
                                                 :adv "too"
                                                 :quant "much" }}}}
  :pred {p adjective
          :pred {p quality
                :topic "good" }}}
```

Figure 3-3. The hierarchical linguistic frame for the sentence: "*Their menu was a good one that didn't try to do too much*." The negation word "*not*" is highlighted, as well as the adjective predicate. The hierarchical structure shows that the negation word is within the adjective clause in the complement sub-frame, and does not scope over the adjective.

With such hierarchical linguistic frames, we could successfully predict the scope of the reference of the negation over the correct constituent of a sentence and create proper association between negation and its modified words. Furthermore, the linguistic parsing approach relies on linguistic features that are independent of word frequencies. Therefore, it can retrieve very rare phrases which are very hard to derive from correlated topic models or frequency statistics (e.g., "*very greasy chicken tikka masala*," "*absolutely delicious spaghetti carbonara*").

3.2 Linear Additive Model for Sentiment Degree Scoring

After extracting context-relevant phrases with linguistic analysis, the next task is to explore a robust general solution for assessing the sentiment values of the extracted phrases. Our goal is to estimate a numerical sentiment degree for each expression on the phrase level. Given a user's spoken input query, the dialogue system needs to understand the sentiment expressed in the user's utterance in order to provide appropriate response. A unified numerical sentiment scale would be easier for the system to intepreate and handle rather than various textual expressions. Thus, in this work, we will explore a simple linear model for sentiment assessment. We will try to obtain reliable estimation from crowdsourcing by utilizing grassroots votings from general users.

Our goal in modeling sentiment is to investigate whether a simple linear correction model can capture the polarity contribution of all adverbials and adjectives. For example, is it appropriate to adjust the orientation level of sentiment for multiple adverbs, including negation, via a linear additive model? That is, can "not very good" be modeled as $not(very(good))$?

To calculate the numerical sentiment values for phrases, there are three major problems to solve: 1) how to associate numerical scores with textual sentiment; 2) whether to calculate sentiment scores for adjectives and adverbs jointly or separately; 3) whether to treat negations as special cases or in the same way as modifying adverbs.

There have been studies on building sentiment lexicons to define the strength of sentiment of words. For example, Esuli and Sebastiani (2006) constructed a lexical resource, SentiWordNet, a WordNet-like lexicon emphasizing sentiment orientation of words and providing numerical scores of how objective, positive and negative these words are. However, lexicon-based methods can be tedious and inefficient and may not be accurate due to the complex cross-relations in dictionaries like WordNet. Instead, our primary approach to sentiment scoring is to make use of community-generated data such as ratings from general users. For example, in product reviews collected from online forums, the format of a review entry often consists of three parts: pros/cons, free-style comment and user rating (as exemplified in Figure 3-1). We assume that the rating by a user is normally *consistent* with the tone of the text published by the same user. By associating the rating with review texts (pros/cons and free-style comment) from each user, we can easily associate numerical scores with textual sentiment.

A simple strategy of rating assignment is to take each extracted adverb-adjective pair as a composite unit. However, this method is likely to lead to a large number of rare combinations, thus suffering from sparse data problems. Therefore, an interesting question to ask is whether it is feasible to assign to each adverb a perturbation score, which adjusts the score of the associated adjective up or down by a fixed scalar value. This approach thus hypothesizes that "*very expensive*" is as much worse than "*expensive*" as "*very romantic*" is better than "*romantic*." This allows us to pool all instances of a given adverb regardless of which adjective it is associated with, in order to compute the absolute value of the perturbation score for that adverb. Therefore, we consider adverbs and adjectives separately when calculating the sentiment score, treating each modifying adverb as a universal quantifier which consistently scales up/down the strength of sentiment for the adjectives it modifies.

Furthermore, instead of treating negations as a special case, the universal model works for all adverbials. The model hypothesizes that "*not bad*" is as much better than "*bad*" as "*not good*" is worse than "*good*," i.e., negations push positive/negative adjectives to the other side of sentiment polarity by a universal scale. This again, allows us to pool all instances of a given negation and compute the absolute value of the perturbation score for that negation, in the same way as dealing with modifying adverbs.

Thus, for each adjective, we collect all the occurrences of this adjective in the corpus, and average all the ratings from each user who published a comment that contains this adjective:

$$Score(adj) = \frac{\sum_{i \in P} \frac{N}{n_{r_i}} \cdot r_i}{\sum_{r_i} \frac{N}{n_{r_i}}} \tag{3.1}$$

where P represents the set of appearances of adjective adj, r_i represents the associated user rating in each appearance of adj, N represents the number of entities (e.g., restaurants, hotels) in the entire data set, and n_{r_i} represents the number of entities with rating r_i. The score is averaged over all the appearances, weighted by the frequency count of each category of rating to remove bias towards any category. Figure 3-4 illustrates the process of generating averaged sentiment scores for adjectives from user-generated comments and ratings. From each user, the adjectives in the “Pros” and “Cons” are associated with the “Overall rating” given by the same user. The ratings on each adjective are then averaged among all the data within the corpus.

As for adverbs, using a slightly modified version of Equation (3.1), we can get an average rating for each *adverb-adjective* pair ($adv - adj$). For each adverb adv, we get a list of all its possible combinations with adjectives. Then, for each adjective adj in the list, we calculate the distance between the rating of $adv - adj$ pair and the rating of the adj alone. We then aggregate the distances among all the pairs of $adv - adj$ and adj in the list, weighted by the frequency count of each $adv - adj$ pair:

$$Score(adv) = \sum_{t \in A} \frac{count(adv, adj_t)}{\sum_{j \in A} count(adv, adj_j)} \cdot Pol(adj_t) \cdot (r(adv, adj_t) - r(adj_t)) \tag{3.2}$$

where $count(adv,\ adj_t)$ represents the count of the combination $adv - adj_t$, A represents the set of adjectives that co-occur with adv, $r(adv, adj_t)$ represents the sentiment rating of the combination $adv - adj_t$, and $r(adj_t)$ represents the sentiment rating of the adjective adj_t alone. $Pol(adj_t)$ represents the polarity of adj_t, which is assigned a value 1 if adj_t is positive, and -1 if negative.

Specifically, negations are well handled by the same scoring strategy, treated exactly the same as modifying adverbs, except that they get such strong negative scores (as shown in the left-most scale in Figure 3-5) that the sentiment of the associated adjectives is pushed to the other side of the polarity scale.

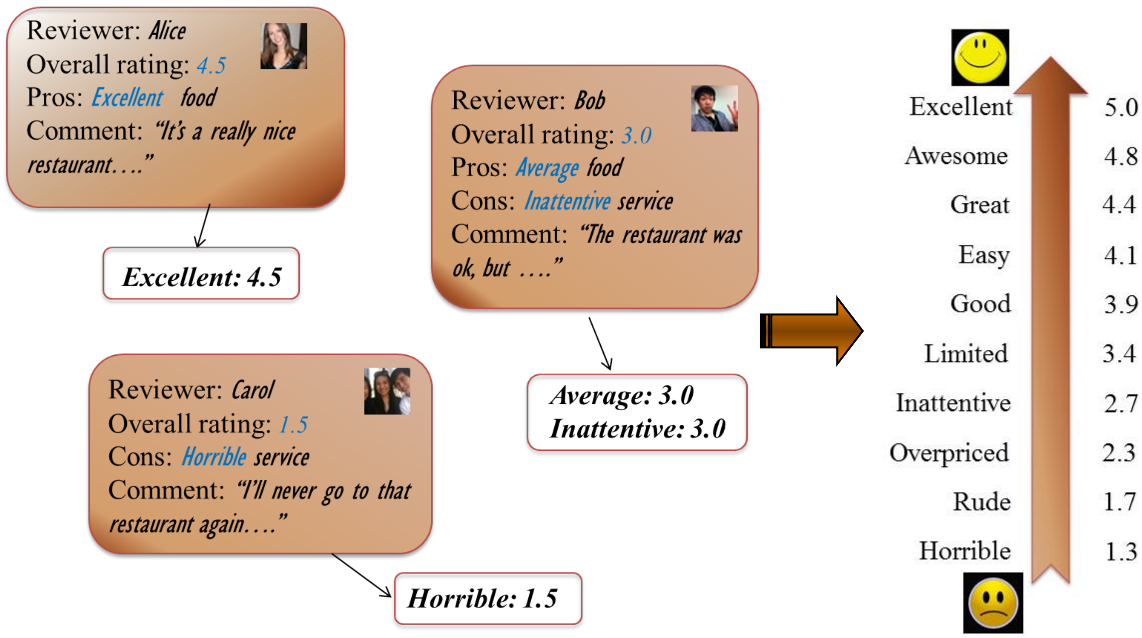

Figure 3-4. Illustration of generating sentiment scores for adjectives. On the left-hand side are original reviews published by different users. On the right-hand side is a scale of adjective sentiment, from positive to negative (top to bottom).

After obtaining the averaged sentiment rating for adjectives and adverbs, we could assign a linearly combined sentiment score as the measurement of sentiment degree to each phrase (*negation-adverb-adjective-noun*) extracted by linguistic analysis, as given by:

$$Score\left(neg\big(adv(adj)\big)\right) = r(adj) + Pol(adj) \cdot r(adv) + Pol(adj) \cdot r(neg) \quad (3.3)$$

where $r(adj)$ represents the rating of adjective adj, $r(adv)$ represents the rating of adverb adv, and $r(neg)$ represents the rating of negation neg. $Pol(adj)$ represents the polarity of adj, which is assigned a value 1 if adj is positive, and -1 if negative. Thus, if

adj is positive, we assign a combined rating $r(adj) + r(adv)$ to this phrase. If it is negative, we assign $r(adj) - r(adv)$. Specifically, if it is a negation case, we further assign a linear offset $r(neg)$ if adj is positive or $-r(neg)$ if adj is negative. Figure 3-5 shows an illustration of the linear additive model for phrase sentiment scoring.

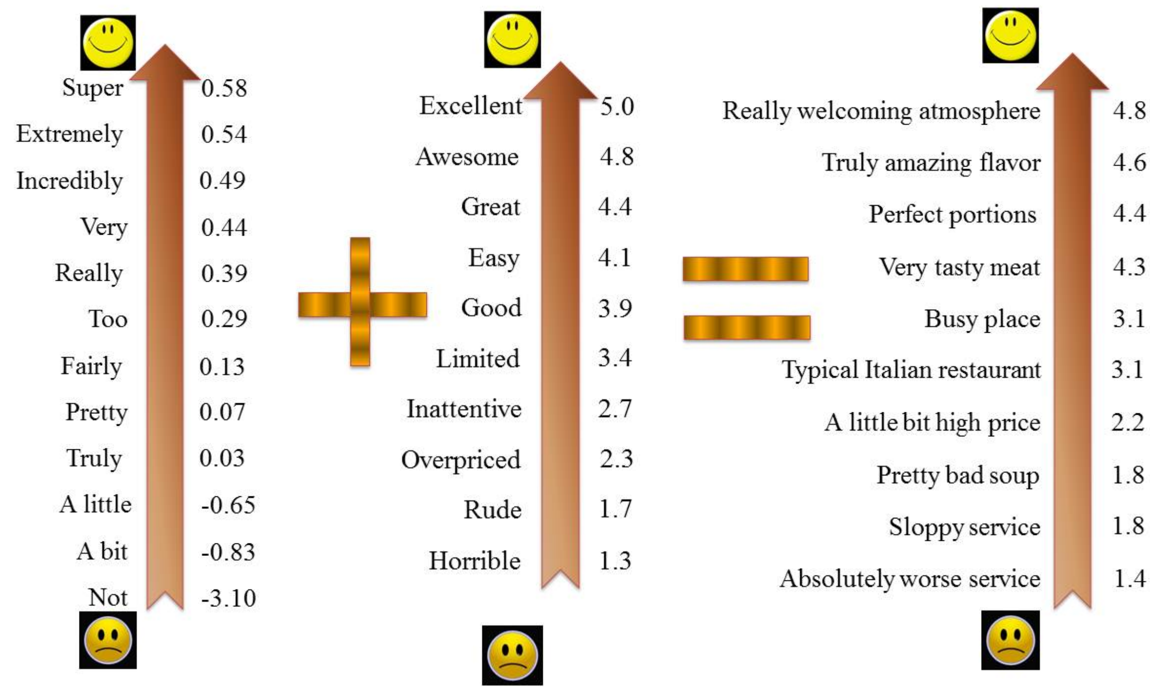

Figure 3-5. Illustration of sentiment computation with the additive model. On the left-hand side are the scale of sentiment strength for adverbs and adjectives. On the right-hand side is the scale of sentiment scores for phrases, positive to negative from top to bottom, obtained by linearly cumulating the sentiment scores of adverbs and adjectives.

3.3 Phrase Classification and Opinion Summary Generation

Given the set of opinion phrases extracted from user-generated data and a sentiment value assigned to each phrase, the next step is to choose the most representative (i.e., informative and relevant) phrases to generate an opinion summary database (Liu et al., 2010a). The task of phrase selection can be defined as a classification problem:

$$y = \bar{\theta} \cdot \bar{x} = \sum_{i=1}^{n} \theta_i x_i \tag{3.4}$$

where y is the label of a phrase, which is assigned a value '1' if the phrase is highly informative and relevant, and '-1' if the phrase is uninformative. $\bar{x}$ is the feature vector extracted from the phrase, and $\bar{\theta}$ is the coefficient vector.

Classification models such as SVMs (Joachims, 1998) and decision trees (Quinlan, 1986) can be trained to automatically classify high/low informative phrases. From each phrase, we extract a set of features for model training. Learning features include standard statistical features (such as unigram/bigram probabilities) and sentiment features (such as sentiment value of the phrase), as well as underlying semantic features (e.g., whether the topic of the phrase fits in a domain-specific ontology). These features are treated as x_i in Equation (3.4) and a classification model can be learned from the training data. Phrases in the test set labeled with "1" by the classification model are considered as highly informative phrases and can be further pruned as well as catalogued to create concise summaries.

Generally speaking, phrases with neutral sentiment are less informative than those with strong sentiment, either positive or negative. For example, "fried seafood appetizer," "baked halibut," "electronic bill" and "red drink" do not indicate whether a restaurant is worth trying, as they did not indicate whether the fried seafood appetizer or the baked halibut are good or bad. Therefore, we take the sentiment score of each phrase generated from the cumulative offset model (as aforementioned in Section 3.2) as a sentiment feature, which shows not only the polarity of sentiment but also the degree of orientation level. We also employ a set of standard statistical features for model training, such as the unigram probability of the adjective in a phrase, the unigram probability of the noun in a phrase, the unigram probability of the phrase and the bigram probability of the adjective-noun pair in a phrase.

Statistical features, however, fail to reveal the underlying semantic meaning of phrases. To capture the semantic importance of each phrase, we first cluster the topics of phrases into generic semantic categories using a language-model based algorithm:

$$\begin{aligned} P(t_c \mid t_i) &= \sum_{a \in A} P(t_c|a) \cdot P(a|t_i) \\ &= \sum_{a \in A} \frac{P(a,t_c)}{P(a)} \cdot \frac{P(a,t_i)}{P(t_i)} \\ &= \frac{1}{P(t_i)} \sum_{a \in A} \frac{1}{P(a)} \cdot P(a,t_c) \cdot P(a,t_i) \end{aligned} \tag{3.5}$$

where A represents the set of all the adjectives in the corpus. We first select a small set of initial topics with the highest frequency counts (e.g., "food," "service" and "atmosphere" in the restaurant domain). Then, for each of the other topics t_c (e.g., "chicken," "waitress" and "décor"), we calculate its similarity with each initial topic t_i based on the adjective-noun bigram statistics. For those topics with conditional probability higher than a threshold for an initial topic t_i, we assign them to the cluster of t_i; assuming intuitively that these topics have high semantic similarity with the cluster topic t_i, given that they co-occur with the same set of adjective set most frequently. We then use this as a semantic feature, e.g., whether the topic of a phrase belongs to a generic semantic category. Table 3-2 gives some topic clustering examples.

Category	**Relevant Topics**
food	*appetizer, beer, bread, fish, fries, ice cream, margaritas, menu, pizza, pasta, rib, roll, sauce, seafood, sandwich, steak, sushi, dessert, cocktail, brunch*
service	*waiter, staff, management, server, hostess, chef, bartender, waitstaff*
atmosphere	*décor, ambiance, music, vibe, setting, environment, crowd*
price	*bill, pricing, prices*

Table 3-2. Topic to semantic category clustering. The column of “Category” represents the initial topics which have the highest frequency in the corpus. The words in the column of “Relevant Topics” are the other topics that are assigned to each category based on the bigram similarity.

This language-model-based method relies on bigram probability statistics and can well cluster highly frequent topics in generic topic categories. Domain-specific categories, however, may contain a very large vocabulary. For example, in the restaurant domain, the category of “food” contains various topics from generic sub-categories (such as “sushi,” “dessert” and “sandwich”) to specific courses (such as “bosc pear bread pudding” and “herb roasted vermont pheasant wine cap mushrooms”). It would be a similar case in other domains. For example, consumer products, movies and books all have domain-independent generic categories (e.g., “price,” “released date”) and domain-specific categories (e.g., technical features of consumer products, casts of movies, and authors of books). These domain-specific topics normally have very low frequencies in a UGC corpus, yet they are very context-relevant and valuable. But many of them are discarded by the frequency-based topic clustering.

To recover these context-relevant yet low-frequency topics, we employ external context resources such as a context-related ontology, which can be constructed from structured web resources such as online menus of restaurants, lists of actors and actresses from movie databases, and specifications of products from online shops. For example, Figure 3-6 shows a partial online restaurant menu, including a few courses listed in two categories: "Entrée" and "Dessert." An example of a structured ontology derived from this menu is shown at the bottom, which successfully includes low-frequency but context-relevant phrases such as "spicy honey-mustard bbq sauce" and "toasted coconut panna cotta." Based on such context-relevant ontology, another set of semantic features covering low-frequency topics can be extracted (e.g., whether a phrase contains the name of a specialty) for the classification model training.

After the classification, phrases identified with positive labels (highly informative and relevant ones) are further clustered into different aspects according to the semantic categories and the hierarchical ontology. An average sentiment score for each aspect is calculated by:

$$ave(s_t) = \frac{\sum_{j \in N_s} r_j}{|N_s|} \tag{3.6}$$

where s_t represents the aspect s of entry t (t can be a restaurant, a movie, or a consumer product), N_s represents the set of phrases in the cluster of aspect s, and r_j represents the sentiment score of phrase j within the cluster.

Entrée
Roasted Pork Loin Wrapped In Bacon with watermelon and red onion salad spicy honey-mustard bbq sauce
Spicy Halibut And Clam Roast with bacon braised greens, white beans and black trumpet mushrooms
Parmesan and Caramelized Shallot Wrapper Style Ravioli turnip greens and white truffle oil
Herb Roasted Vermont Pheasant Wine Cap Mushrooms, Pearl Onions and Fava Beans
Dessert
Chocolate Tasting Plate of white chocolate bombe milk chocolate creme brulée and dark chocolate flourless cake
White Fruit Tasting Plate of warm apple strudel butterscotch, Bosc Pear bread pudding and toasted coconut panna cotta

Entrée	*roasted pork loin*
	red onion salad
	spicy honey-mustard bbq sauce
	caramelized shallot wrapper style ravioli
	herb roasted vermont pheasant wine cap mushrooms
Dessert	*chocolate tasting plate*
	white chocolate bombe milk chocolate crème brulee
	dark chocolate flourless cake
	white fruit tasting plate
	warm apple strudel butterscotch
	bosc pear bread pudding
	toasted coconut panna cotta

Figure 3-6. Example of a partial online menu and an exemplary ontology derived. The simplified online menu is shown on the top, with two categories: “Entrée” and “Dessert.” The structured ontology derived from this menu is shown at the bottom.

The opinion-related phrases are extracted from a large number of documents, and many of them may include the same topic (e.g., “*good fish*,” “*not bad fish*” and “*above-average fish*” for one restaurant). Thus, redundancy elimination is required. In each category, among those phrases with the same topic, we select the phrase whose sentiment score is closest to the average score of this aspect as the most representative phrase:

$$j^* = argmin_{j \in N_i}(|r_j - ave(s_t)|) \quad (3.7)$$

where $ave(s_t)$ represents the average sentiment score of aspect s, N_i represents the set of phrases on the same topic i, and r_j represents the sentiment score of phrase j within N_i. The goal is to find the phrase j^* for each topic i, the sentiment score of which has the smallest distance to the average aspect rating.

This sequence of phrase classification, topic categorization, phrase pruning and redundancy elimination results in a summary database. An example database entry is exemplified in Table 3-3, which contains lists of descriptive phrases ("*:atmosphere*," "*:food*," "*:service*," and "*general*") as well as aspect ratings ("*:atmosphere_rating*," "*:food_rating*," "*:service_rating*," *and* "*general_rating*").

Name	*"Devlin's restaurant and bar"*
City	*"Brighton"*
Cuisine	*"American"*
Atmosphere	*"romantic date" "elegant decor"*
General	*"great place"*
Food	*"wonderful martinis" "good wine" "great fish"*
Service	*"fast service"*
Specialty	*"martinis" "wine" "fish"*
Atmosphere_rating	*"4.2"*
General_rating	*"4.2"*
Food_rating	*"4.3"*
Service_rating	*"3.9"*

Table 3-3. Example of an opinion summary database in the restaurant domain generated by the proposed approach, including both catalogued representative phrases and aspect ratings.

3.4 Chapter Summary

In summary, to process unstructured user-generated content, we proposed a linguistic parse-and-paraphrase paradigm to extract representative context-relevant phrases utilizing the "linguistic frame," which preserves linguistic structure of a sentence by encoding different layers of semantic dependencies. This allows us to employ more sophisticated high-level linguistic features (e.g., long distance semantic dependencies) for phrase extraction. The proposed approach makes use of a well-formed probabilistic syntactic grammar, and negations are treated appropriately based on the hierarchy structure of an utterance; thus, context-relevant phrases can be extracted reliably.

We also developed an accumulative linear offset model for sentiment assessment, which treats negations in the exact same way as modifying adverbs. This yields a very generic and straightforward solution to modeling the degree of sentiment expressions. The proposed sentiment prediction model takes modifying adverbs and negations as universal scales on strength of sentiment, and conducts cumulative calculation on the sentiment value for the associated adjectives. With this model, we can assess not only the sentiment polarity (positive or negative) of an opinion segment, but also a numerical degree of sentiment orientation in a measurable way. The sentiment scores of adjectives and adverbs are averaged over ratings from a large number of general users. Thus, the sentiment scales are, although data-dependent, considered as reliable and close to human judgment.

To create condensed opinion summaries, we applied a phrase classification model to choose highly informative and context-relevant phrases from the original set of extracted phrases. The classification model can be trained on statistical, semantic and sentiment features. Topic clustering and pruning are further applied to create aspect ratings and catalogued summaries for dialogue purposes.

For evaluation purposes, we implement the proposed linguistic-parsing approach on a real corpus of user-generated content, and the experiments will be described in Chapter 5 (Section 5.2). Experimental results will show that the proposed parse-and-paraphrase paradigm can make use of linguistic features in a combination with statistical features and outperform standard phrase extraction baselines. We will also apply the proposed sentiment assessment method to the same corpus, and experiments on comparing our automatic sentiment assessment system to human judgment will be described in Chapter 5 (Section 5.3). Section 5.4 will describe the evaluation of the phrase classification approach compared to human annotations as well.

Chapter 4

Dialogue Modeling and Response Generation

This thesis aims to conduct research on a universal framework for developing dialogue systems that can harvest user-generated content and present the summarized information via speech-based human-computer interaction. In the previous chapters, we explored how to interpret users' intention from spoken utterances, and how to process and summarize unstructured data. We investigated approaches to extracting the essence of unstructured user-generated content and presenting it in a structured representation (as exemplified in Figure 3-7).

To make the system present the highlighted information to users in an interactive dialogue, a dialogue modeling mechanism driven by the UGC summary database is required to handle discourse and dialogue. In this chapter, we will explore how to develop a speech-based interface to help users access summarized UGC data efficiently. To be consistent, we will continue using the example in Figure 3-7 for demonstration. An example of a spoken conversation between a user and a dialogue system based on the example database is shown in Figure 4-1.

U: Are there any American restaurants in Brighton?

S: There are 61 American restaurants in Brighton. All of the rated restaurants have good or excellent ambiance. Many are on Washington Street, Commonwealth Avenue and Market Street.

U: Show me the ones that have nice ambiance.

S: I found 4 entries. Devlin's Restaurant and Bar is perfect for romantic date, Stockyard Restaurant has friendly atmosphere, Tasca Tapas Restaurant has excellent ambiance, and Wings Express has friendly vibes.

U: Which one has good martinis?

S: Devlin's Restaurant and Bar serves American food. It has wonderful martinis, good wine, great fish. It is perfect for romantic date. Great place.

Figure 4-1. Example of a conversation between a user and a dialogue system, which provides the user with information about summarized user-published reviews in the restaurant domain ("U" is the user and "S" is the system).

4.1 Dialogue Modeling

To respond appropriately to users' request, a system should be able to identify the scope of users' questions. To gain intuition on what kinds of questions real users may ask when using a speech-based inquiry system, we conducted a user study (Liu et al., 2010b) by collecting sentences from general users regarding restaurant inquiries. From the data collection we observed that most of the users' queries fall into one of two categories: *feature-specific* questions and *qualitative* questions. For example, some users were looking for a particular restaurant that serves "mango-peach sangria," or "kobe beef sushi," or "spaghetti cuttlefish ink," or other house specialties. And other users were asking about the general quality of restaurants, such as general food quality or ambiance.

This can be generalized to other domains as well. For example, in a shopping domain, a user may want to buy a digital camera with specific features, such as with image stabilization or with 10 megapixels. Users may also ask about the general quality of products, such as customer service or low price. We call the first type of queries as "feature-specific" queries, and the second type as generic "qualitative" queries.

To support a speech-based UGC access interface, a system is expected to understand the user' spoken query and assist the user by providing the correct information that the user inquired about. Thus, we will explore how to make a dialogue system handle both general and specific queries.

4.1.1 Feature-Specific Entity Search

As aforementioned in Chapter 3, we used linguistic parsing techniques to extract context-relevant phrases from user-generated data corpus. Here, to allow the system to identify feature-related topics in users' utterances, we extract all the nouns and noun phrases (based on parsing features) from the set of extracted context-relevant phrases as domain-specific topics. Then, we modify the context-free grammar used for parsing users' utterances by including these feature-specific topics as a *word class*. When a feature-specific query utterance is submitted by a user, the linguistic parser will generate a hierarchical structure -- a linguistic frame -- for the utterance, which encodes the syntactic and semantic structure of the utterance and, especially, identifies the feature-related topics in the specific word class.

Figure 4-2 demonstrates the procedure of interpreting a user's utterance, extracting the meaning representation and searching the UGC database. The user submitted a feature-specific query: “Are there any restaurants in Brighton that have *good martinis*?” A set of key-value pairs can be extracted from the parsing result (the second step of Figure 4-2) representing the semantic meaning of the utterance. The feature-specific topic (“*specialty: martinis*”) can be identified during the parsing process as “martinis” was in the list of domain-relevant topics extracted from the UGC corpus and was included in the word class “specialty” in the context-free grammar.

With the key-value pairs, the system can filter the database by matching the inquired features (*specialty* = “*martinis*”) with the UGC database. Those database entries which satisfy the constraints will be retrieved as the candidates (as exemplified in the last step of Figure 4-2), and a response utterance can be generated based on the search result.

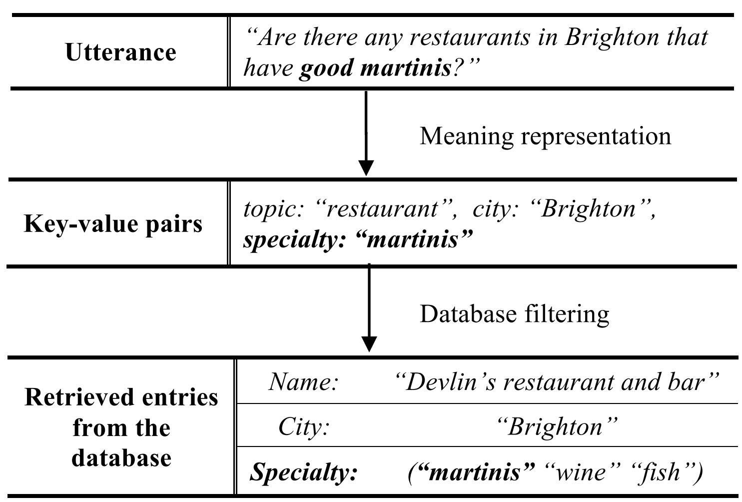

Figure 4-2. Illustration of the procedure of feature-specific entity search. Given a user’s utterance, a list of key-value pairs including the “feature-specific” topic is extracted from the parsing results as the meaning representation. These key-value pairs are used as the database filters to retrieve the database entries that match the query.

4.1.2 Qualitative Entity Search

The feature-specific queries can be handled well with keyword search (e.g., search by "martinis," "sushi," "fish"). For high-level qualitative questions (e.g., "Show me some American restaurants with *nice ambience*"), however, the keyword search method is problematic, as there are normally multiple variants of expressions with the same qualitative meaning. For example, given the query "*nice ambience*," entities with "*friendly vibes*" or "*excellent atmosphere*" also satisfy the query and should be retrieved, but they would have been missed by keyword search methods due to different expressions from the query words.

To keep the richness and varieties of the original UGC data, a better solution is to enable a dialogue system to cluster similar topics into categories as well as identify sentiment strength of various expressions. As aforementioned in Chapter 3, we proposed a method for calculating a sentiment score for each opinion-expressing adjective and adverb (e.g., "*bad: 1.5*," "*good: 3.5*," "*great: 4.0*," on a scale of 1 to 5). Here, we make use of these sentiment scores to convert the qualitative queries into measurable values. These numerical sentiment values can be used to search the database on aspect ratings for general categories. In this way, opinion expressions can be distinguished by a measureable scale and database entries with descriptive words different from the user's query, but with similar sentiment values, can be recovered.

Figure 4-3 shows an exemplified procedure of a dialogue system handling generic qualitative queries. Similar to the case of feature-specific search, when a user's utterance (e.g., "Show me some American restaurants with *great food*") is submitted to the system and passed through speech recognition, a linguistic parser parses the sentence into a linguistic frame, from which a set of key-value pairs (the second step in Figure 4-3) is extracted as a meaning representation of the utterance. However, in this case, instead of using the original key-value pairs as database filters, we use a converted numerical measurement to present the qualitative query.

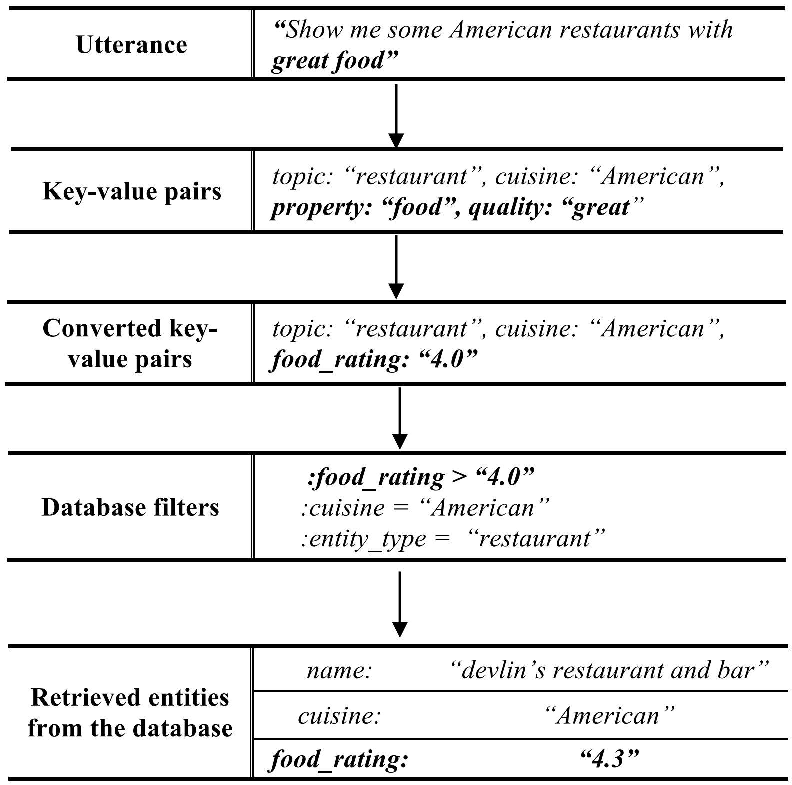

Figure 4-3. Illustration of the procedure of qualitative entity search. Given a user's utterance, a list of key-value pairs is extracted from the parsing results as the meaning representation. The sentiment-related key-value pairs are converted to measurable sentiment values, which are used as database filters to retrieve qualified database entries.

As shown in Figure 4-3, by mapping the descriptive word "great" into its sentiment score "4.0" (as described in Chapter 3, the sentiment scores for descriptive words are learned automatically from user-generated data. Here, we just use "4.0" as an arbitrary example), the key-value pairs "*property: food, quality: great*" are converted to "*food_rating: 4.0*" (the third step of Figure 4-3). An algorithm can be defined as filtering the database for entities that have scores higher than the inquired value (e.g., "*:food_rating > 4.0*"), which indicates that those expressions with stronger sentiment than the user's inquiry satisfy the requirement. In this way, the qualitative query can be easily converted to measureable values; and the entities that are in the same range of sentiment degree as the user's query can be retrieved from the database (as exemplified in the last step of Figure 4-3).

4.2 Probabilistic Language Generation

After retrieving the search results from the database, the next step is to present the information in a natural dialogue, i.e., to encode the retrieved database entries into natural language utterances. In this task, each database entry contains one or a few lists of phrases as a description summary (e.g., "romantic date, elegant decor"). The challenge, therefore, is how to choose an appropriate *predicate* for each phrase, in order to chain them into a natural sentence (as illustrated in Figure 4-4).

Here, we inherit the generic-domain language generation engine by Baptist and Seneff (2000). The template-based approach is robust and reliable. However, manually pre-defining templates for each specific linguistic pattern is tedious and not scalable. To avoid the human effort involved in the predicate selection task, we propose a corpus-based approach to automatically learn predicate-topic association from a large user-generated corpus based on linguistic parsing statistics.

Retrieved database entry	*Name:*	*"Jonny D's"*
	General:	*"great place"*
	Atmosphere:	*"nice jazz music, best breakfast spot, great vibes"*

↓

Generated response utterances	*"Jonny D's is a great place. The restaurant has nice jazz music and great vibes. It is the best breakfast spot."*

Figure 4-4. Example of response utterances generated from a database entry. The input is a summary database entry. The goal is to encode the phrase-based information into a string of utterances, i.e., for each phrase in the catalogued descriptions, how to automatically choose a predicate that best matches the topic of the phrase.

The proposed approach consists of four stages: 1) plant seeds (i.e., words of interest) in the context-free grammar; 2) identify semantic structures associated with the seeds; 3) extract association pairs of linguistic patterns and the seeds; and 4) calculate the probabilities of the occurrences of all the association pairs. First, we collect a set of words as the seeds. For this particular task, we extract all the nouns and noun phrases that occur in the descriptive summaries as the seeds; but in other tasks and other domains any set of words can be selected. As aforementioned, the system could use a context-free grammar to parse each sentence into a linguistic frame, which can be further paraphrased into a meaning representation. To identify the topics for which we want to choose predicates, we modify the grammar rules by giving these topic words a specific tag. For example, as shown in the first stage of Figure 4-5, the topics ("vibes," "jazz music" and "breakfast spot") will be assigned with an "**active**" tag in the parsing process, such that the parser can identify these seed words as the "*active*" topics when generating the linguistic frame.

Stage 1: plant "active" topic seeds (partial)

*"vibes" :*active* "1"*
*"jazz music" :*active* "1"*
*"breakfast spot" :*active* "1"*

Stage 2: identify predicates and clauses associated with the seeds

clause_template <default template rules>
*($set :*active* :*active*[:complement])*
*($set :*active* :*active*[:subject])*

predicate_template <default template rules>
*($set :*active* :*active*[:object])*

Figure 4-5. Examples of grammar rules for identifying topic seeds and the associated patterns. Stage 1 shows a few "active" topic seeds that are planted in the grammar rules. Stage 2 shows a few generation rules for identifying the patterns (clauses and predicates) that are associated with these "active" topics. Based on these grammar rules, a corpus will be parsed and the "active" topics occurring in each sentence of the corpus will be identified and extracted along with the corresponding predicates and clauses.

The second stage is to identify all the linguistic patterns associated with each seed. In a preprocessing step, we assign identifiable tags to the predicates and clause structures that are semantically related to the seeds. As shown in the second step of Figure 4-5, the first rule is a clause-based rule. The "*<default template rules>*" represent some grammar rules used to rewrite a sub-parsing result of a clause into a sub-linguistic-frame. The two following sub-rules with "*$set *active**" tags mean that, if the complement or the subject of the clause is an "active" topic, assign an "**active**" tag to the clause as well. The second set of predicate-based rules has similar logic. In this way, when examining syntactic hierarchy of sentences, the system can encode all the linguistic patterns of clauses or predicate-topic relationships that are associated with the topic seeds with special tags (e.g., "**active**").

These association pairs of linguistic patterns and topics can be extracted with customized generation rules from the parsing results. Generation is applied only to the constituents that are marked as "**active**," such that only patterns associated with the seeds are identified. Based on the modified grammar rules, we can learn common linguistic patterns (e.g., predicate-topic associations) from a large corpus. Sentences in the corpus can be parsed with the modified grammar and the "**active**" topics occurring in each sentence of the corpus will be identified and extracted along with the corresponding predicates and clauses.

Given all the association pairs extracted from the corpus, we can calculate the probability of each association pair by:

$$prob\left(pattern_j \middle| seed_k\right) = \frac{count(pattern_j, seed_k)}{\sum_i count(pattern_i, seed_k)} \tag{4.1}$$

where $seed_k$ is a seed word, and $pattern_i$ is every linguistic pattern associated with $seed_k$. The probability of $pattern_j$ for $seed_k$ is the percentage of the occurrences of this pattern among all the occurrences of $seed_k$ in the corpus. This is similar to a bigram language model. A major difference is that the linguistic pattern is not necessarily the previous word before the seed topic. It can be multiple words in a semantic chunk, and it can be a long distance from the seed topic. The long distance semantic relationships are captured by the linguistic parser and its hierarchical encoding structure; thus, it is more reliable than pure co-occurrence statistics or bigrams. An example of a partial probability table learned from a corpus is shown in Table 4-1.

Constituent	**Association pair <predicate: topic>**	**Probability**
PP	*<“at” : “breakfast spot”>*	*0.07*
Clause	*<“is” : “breakfast spot”>*	*0.57*
PP	*<“for” : “breakfast spot”>*	*0.14*
VP	*<“love” : “jazz music”>*	*0.08*
VP	*<“have” : “jazz music”>*	*0.23*
VP	*<“enjoy”: “jazz music>*	*0.08*

Table 4-1. Partial table of probabilities of predicate-topic association pairs (VP: verb phrase; PP: preposition phrase). The column of “Association pair” shows the pairs of associated predicate and topic discovered from the corpus. The column of “Constituent” represents the clause or phrase from which each association pair was extracted. The column of “Probability” shows the probability of predicate association for each topic calculated over the entire corpus.

Given these association probabilities, we can define algorithms to select pairs of topics and patterns. For example, a language generation system can choose the predicate associated with each topic with the *highest* probability to generate the corresponding utterances.

4.3 Chapter Summary

In this chapter we investigated the dialogue modeling and response generation strategies for a speech-based UGC access system. We proposed a set of UGC-database search algorithms to enable a dialogue system to handle both feature-specific and qualitative queries. To handle high-level sentiment-involved questions, we make use of the sentiment scoring strategy proposed in this work to convert qualitative queries into measureable values, which is more intuitive for a system to interpret and handle.

For response generation, a corpus-based approach was proposed to automatically choose predicates for various topics and to create natural sentences based on linguistic parsing statistics. The only domain-dependent part of this approach is the selection of the seeds. The other steps all depend on generic linguistic structures and are domain-independent. Thus, this probabilistic method can be easily applied to generic domains for customizing language generation.

To evaluate the proposed dialogue modeling mechanism as well as the corpus-based method, we apply them to a spoken dialogue system enhanced with a UGC summary database created by previously-described unstructured data processing methods. The implementation of the dialogue system and the evaluations with real users will be explained in Chapter 5 (Section 5.5). The user study will show that the proposed approaches can make a dialogue system generate reliable and natural conversations to help general users obtain UGC information efficiently.

Chapter 5

Experiments and Evaluation

In this chapter, we will describe our experiments on each proposed approach described in previous chapters with real data from several domains. A restaurant-domain spoken dialogue system for user-generated content sharing is implemented as a prototype system, and an evaluation of the system with real users is also provided.

5.1 Query Interpretation

To evaluate the proposed approach to lexicon modeling for query understanding, we conducted experiments on textual queries formulated in both natural language and keywords in three domains: *Restaurant*, *Hotel* and *Movie*. The statistics on the training and test sets in each domain are shown in Table 5-1. We asked human annotators to manually label the data. The annotators independently segmented each query into slots and assigned each slot a semantic class label selected from Table 5-2. Segments that do not belong to any of the semantic classes are assigned an "Other" label.

Domain	Training set		Test set		Number of entries in the baseline lexicon
	#Queries	#Slots	#Queries	#Slots	
Restaurant	2340	5269	601	1331	~500k
Hotel	1572	3729	406	992	~90k
Movies	1768	2257	540	654	~120k

Table 5-1. Statistics on training/test sets in three domains. The column of "#Queries" shows the number of natural language or keyword queries in the training/test set in each domain, and the column of "#Slots" shows the total number of slots labeled by the annotators on all the queries in the training/test set in each domain.

Domain	Semantic classes
Restaurant	*cuisine, restaurant type, amenities, menu item, restaurant name, described as, location, opening hour, star rating, price range, reservation date, reservation time, reservation party size, meal type*
Hotel	*hotel type, hotel name, location, room type, adult number, child number, reward program, smoking, checkin date, checkout date, nights, number of rooms, star rating, described as, price range, amenities*
Movie	*movie type, character, award, movie name, location, theater, date, release date, time, star rating, mpaa rating, genre, nationality, director, review site, year, language, star, number of tickets*

Table 5-2. Semantic classes defined for each domain in the annotated data. There are in total 14 classes in the "restaurant" domain, 16 classes in the "hotel" domain, and 19 classes in the "movie" domain.

The evaluation metrics are precision, recall and F1 (the harmonic mean of precision and recall) at the slot level, excluding “Other” slots. A slot is considered as identified correctly if and only if it is segmented correctly and tagged with the same label as annotated. We used a semi-Markov CRF model as our baseline. Transition features, lexical features and semantic features were used for model training (Li, 2010). For semantic features, we used exact match on baseline lexicons, which were obtained from databases for hotels, restaurants, and movies. The baseline lexicons we applied our approaches to are *HotelName* (~90k entries), *RestaurantName* (~500k entries) and *MovieTitle* (~120k entries).

To implement our lexicon expansion and weighting approach, we used query logs collected over a year from the commercial search engine Bing[4]. To learn new lexicons with generative models, we extracted lexicon candidates using the pattern “imdb.com/title” from URLs in query logs for the *MovieTitle* entity; we also extracted lexicon candidates from titles of documents in query logs using 34 most frequent keywords (e.g., “bed and breakfast”) in the hotel domain for *HotelName* and 45 keywords (e.g., “steakhouse”) in the restaurant domain for *RestaurantName*. These keywords are learned automatically from the query logs based on frequency statistics. For discriminative models, we employed a maximum entropy classifier as the context classification model and used *n*-grams as features.

[4] This work was done at Microsoft Research in collaboration with Xiao Li, Alex Acero and Ye-Yi Wang. The data and the code belong to Microsoft. Thus we did not use the data set for other experiments, nor did we incorporate the code in our prototype system. Instead, we use public user-generated content on the Web for the evaluation through Section 5.2 to Section 5.5, and we use the NLU component developed in our group to implement the prototype system. But the semantic tagging based language understanding approach can be potentially applied to dialogue systems.

Experimental results on generative and discriminative models (Liu et al., 2011a) are given in Table 5-3. Figures 5-1, 5-2 and 5-3 show the performance on each domain, respectively. "*P*" is precision and "*R*" is recall, both presented in percentage. "*BS*" denotes the baseline which uses exact match on the baseline lexicons. "*FM*" denotes fuzzy match on baseline lexicons, and "*LE*" denotes lexicon expansion, i.e., exact match on the expanded lexicon learned from the generative model. "*GLW*" denotes lexicon weighting by generative models and "*DLW*" denotes lexicon weighting by discriminative models, both using the same expanded lexicons. And "*FME*" denotes fuzzy match on expanded lexicons.

	Hotel			**Restaurant**			**Movie**		
	P	R	F1	P	R	F1	P	R	F1
BS	86.3	87.3	86.8	85.8	88.1	86.9	78.8	76.9	77.9
BS+FM	87.5	87.6	87.6	85.9	88.4	87.2	83.5	79.5	81.4
BS+LE	87.2	87.4	87.3	85.6	88.2	86.8	81.2	78.4	79.8
BS+LE+GLW	87.4	87.6	87.5	86.0	88.5	87.2	81.1	78.9	80.0
BS+LE+DLW	86.5	87.4	87.0	85.9	88.4	87.1	80.7	78.0	79.3
BS+FM+LE	88.7	87.9	88.3	86.4	88.9	87.6	84.1	80.7	82.4
BS+FM+LE+GLW	89.4	88.6	89.0	86.7	89.1	87.9	84.7	81.2	82.9
BS+FM+LE+DLW	88.4	88.2	88.3	86.3	88.6	87.4	83.8	80.7	82.2
BS+FM+LE+GLW+FME	89.9	88.9	89.4	86.0	88.4	87.2	84.6	81.7	83.1
BS+FM+LE+DLW+FME	89.9	89.0	89.5	86.3	88.6	87.4	84.0	81.2	82.6

Table 5-3. Semantic tagging performance on the restaurant, hotel and movie domains, using different feature combinations of fuzzy match, lexicon expansion and lexicon weighting. Measure criteria are precision ("P"), recall ("R") and F1.

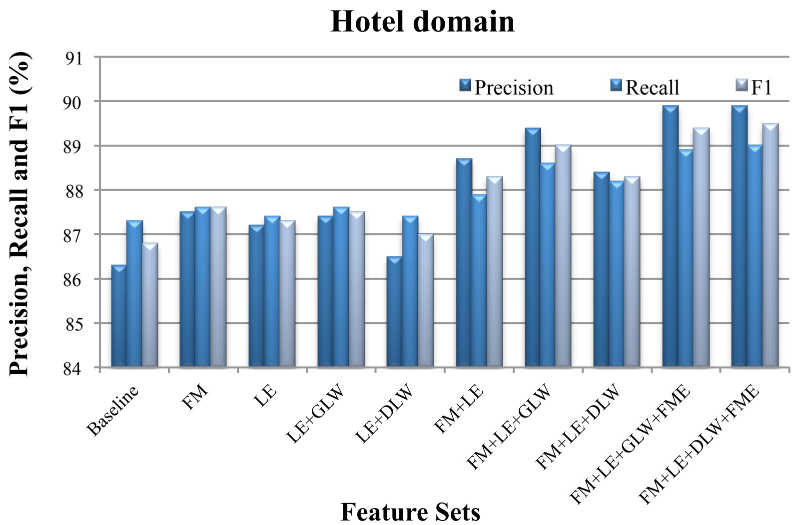

Figure 5-1. Performance of semantic tagging using different feature sets on *Hotel* domain. The best performance (highest F1) was achieved by the feature set of "BS+FM+LE+DLW+FME."

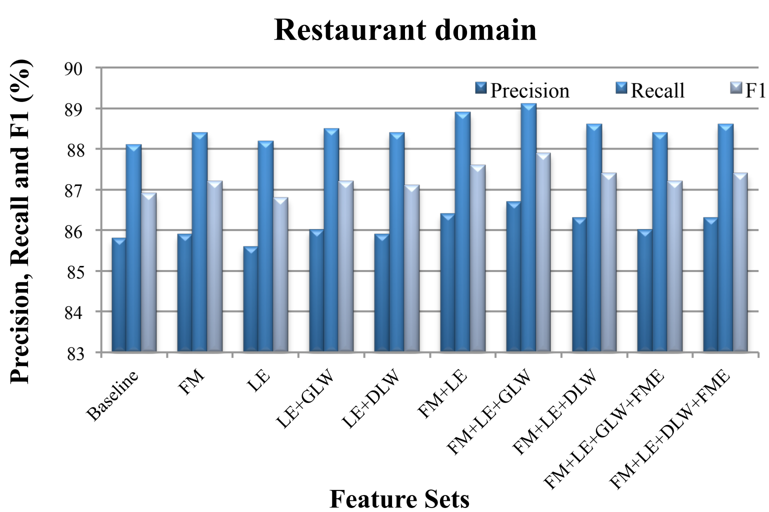

Figure 5-2. Performance of semantic tagging using different feature sets on *Restaurant* domain. The best performance (highest F1) was achieved by the feature set of "BS+FM+LE+GLW".

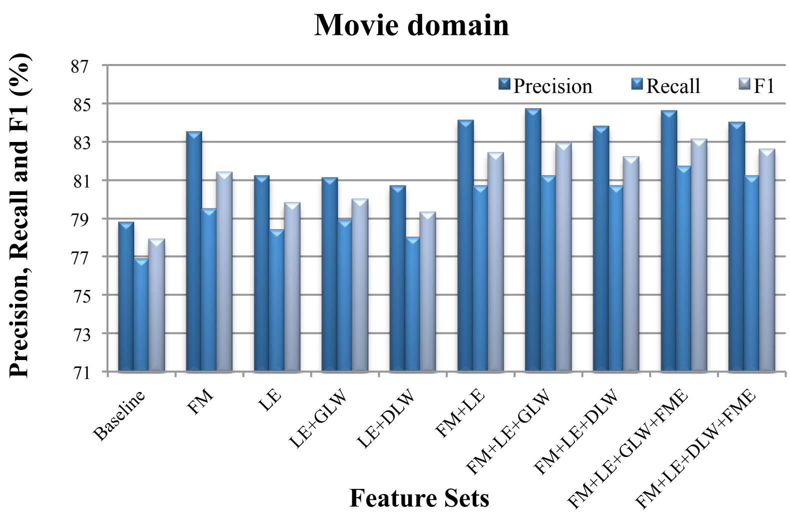

Figure 5-3. Performance of semantic tagging using different feature sets on *Movie* domain. The best performance (highest F1) was achieved by the feature set of "BS+FM+LE+GLW+FME."

Experiments show that, in the hotel and movie domains, each technique (*FM – fuzzy match*, *LE – lexicon expansion*, and *LW – lexicon weighting*) helps to improve the baseline, and the best performance is obtained by combining all techniques in both generative and discriminative models. In the restaurant domain, the best performance on generative models is without *FME* and that on discriminative model is without *FME* and *LW*. The relative improvements in the movie and hotel domains are more than that in the restaurant domain. This is due to the coverage of pre-collected lexicons. In our data, the size of the pre-collected lexicon in the restaurant domain (~500k) is much larger than that in the hotel and movie domains (~90-120k). The expanded lexicons in the hotel and movie domains are both larger than the pre-collected lexicons, while in the restaurant domain the ratio of the expanded lexicon over the pre-collected one is only 0.17. As it is a lexicon-expansion-based approach, the improvement over a smaller pre-collected lexicon (or with a higher ratio of expansion) is expected to be more significant than that over a larger pre-collected lexicon.

The performance obtained using lexicon weights learned from generative models is comparable with that from discriminative models. This shows that taking both popularity and ambiguity into account in lexicon weighting helped semantic tagging, although no one model seems to be significantly better than the other. In future work, we will explore how to combine these two types of features for better performance.

5.2 Linguistic Parsing for Phrase Extraction

In this section, we present a systematic evaluation of the proposed linguistic phrase extraction approach with real user-generated data. We took the restaurant review domain as an example and harvested a collection of 137,569 user-published reviews on 24,043 restaurants in 9 cities in the U.S. from an online restaurant evaluation website[5]. Most of the reviews have both pros/cons and free-style text. An example of reviews from the website was shown earlier in Chapter 3 (Figure 3-1). The pros/cons of a review entry often contain short and well-structured phrases, and have better parsing quality than the long and complex sentences in free-style texts. Thus, for the purpose of evaluation, we take those reviews containing pros/cons as the experimental set, which is 72.7% (99,147 reviews) of the original set, so as to use the pros/cons of reviews as the ground truth.

First, we conducted a pre-processing of sentence-level data filtering. Review data published by general users is often in free-style, and a large fraction of the data is either ill-formed or not relevant to the task. We classified these as out of domain sentences. To

[5] http://www.citysearch.com

filter out such noisy data, we calculated unigram statistics on the corpus and collected high frequency adjectives and nouns as context-relevant or opinion-related words. Any sentence that contained none of these high-frequency nouns or adjectives was rejected from further analysis (e.g., an opinion-free sentence: "Last Friday I went to this place with some friends to celebrate my birthday"). The remaining in-domain sentences were subjected to the second stage, parse analysis and semantic understanding, for topic extraction. Among the experimental set, a set of 857,466 in-domain sentences (67.5%) remained after the sentence filtering process. This set was then subjected to parse analysis (Seneff, 1992a), and 78.6% of them were parsable. Given the parsing results in the format of linguistic frame, we used a set of language generation rules to extract relevant adverb-adjective-noun phrases.

To evaluate the performance of the proposed approach (*LING*) to phrase extraction, we compared it with a baseline method similar to (Hu and Liu, 2004a, 2004b; Liu et al., 2005). We performed part-of-speech tagging on both parsable and unparsable sentences, extracted each pair of noun and adjective that has the smallest proximity, and filtered out those with low frequency counts. Adverbs and negation words that are adjacent to the identified adjectives were also extracted along with the adjective-noun pairs. We call this the "neighbor baseline" (*NB*).

The proposed method is unable to make use of the non-parsable sentences, which make up over 20% of the data. Hence, it seems plausible to utilize a back-off mechanism for these sentences via a combined system (*COMB*) incorporating *NB* only for the sentences that fail to parse.

The phrases in the pros/cons of each review are considered as the ground truth. Performance was evaluated in terms of recall (percentage of phrases in the ground truth that are also identified from the review body) and precision (percentage of extracted phrases by the system that are also in the ground truth). These measures were computed separately for each review, and then averaged over all reviews.

As shown in Figure 5-4 and Table 5-4, the *LING* approach gets both higher recall and higher precision than the *NB* baseline. The *COMB* approach gets the highest recall, with a 4.9% and 17.5% increase from the *LING* approach and the *NB* baseline, respectively. The precision is quite close to that of the *LING* approach (60.8% vs. 61.1%). This shows that the linguistic parsing approach can retrieve more context-relevant phrases by preserving the hierarchical semantic structure of a sentence; and, by combining a keyword matching method for unparsable sentences, the approach can get higher coverage.

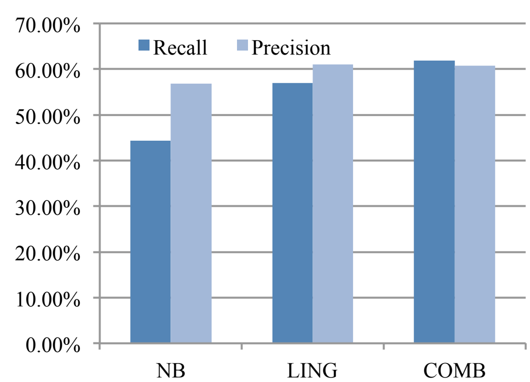

Figure 5-4. Phrase extraction performance on the baseline (*NB*), the proposed *LING* approach and the combined system (*COMB*).

	NB	*LING*	*COMB*
Recall	44.4%	57.0%	61.9%
Precision	56.8%	61.1%	60.8%

Table 5-4. Experimental results of phrase extraction by the *NB* baseline, the proposed *LING* approach and a combined system (*COMB*). The *COMB* system achieves the highest recall and a precision comparable to that of the *LING* approach, both outperforming the baseline.

As shown in the results, the best-preforming system could achieve a precision up to 60%. We suspected that the over-generated phrases (the 40% of phrases that find no mappings in the pros/cons) might not really be a problem. To test this hypothesis, we selected 100 reviews for their high density of extracted phrases, and manually evaluated all the over-generated phrases. We found that over 80% were well formed, correct, and informative. Therefore, a lower precision here does not necessarily mean poor performance, but instead shows that the pros/cons provided by users are often incomplete. By extracting phrases from free-style review texts we can recover additional valuable information at the expense of additional processing.

5.3 Sentiment Analysis

In this section, we evaluate the sentiment scoring approach with the same restaurant-review corpus used in the phrase extraction experiments (in Section 5.2). The pros/cons in a review entry often have clear sentiment orientations. Thus, we use pros/cons to estimate the sentiment values of adjectives, which requires strong polarity association. On the other hand, the frequencies of adverbs in free-style texts are much higher than those in pros/cons, as pros/cons mostly contain adjective-noun patterns. Thus, we used free-style texts instead of pros/cons to calculate the sentiment strength of adverbs.

To obtain reliable ratings, we arbitrarily associated the adjectives in the “pros” of review entries that have a user rating of 4 or 5, and associated the adjectives in the “cons” of review entries with user ratings of 1 or 2 (on a scale of user rating from 1 to 5). Reviews with rating 3 express neutral sentiment, so we associated both “pros” and “cons” with the overall rating in these cases.

Using the algorithms for sentiment scoring (Equation 3-1 and 3-2), we built a mapping table of sentiment scores for adjectives and one of strength scores for common adverbs (Liu and Seneff, 2009). The polarity of sentiment as well as the degree of polarity of an adjective can be distinguished by its score: the higher the sentiment score is, the more positive the adjective is. The complete mapping table for adjectives learned from our UGC corpus can be found in Appendix A.

Table 5-5 gives the strength scores for most common adverbs learned from the review corpus. The higher the strength score is, the more the adverb scales up/down the degree of sentiment of the adjective it modifies. While “not” gets a strong negative score, some adverbs such as “a little” (-0.65) and “a bit” (-0.83) also get negative scores, indicating slightly less sentiment for the associated adjectives.

To evaluate the performance of sentiment scoring, we randomly selected a subset of 1,000 adjective-noun phrases from the set of phrases extracted by our linguistic analysis and asked two annotators to independently rate the sentiment of each phrase on a scale of 1 to 5. We compared the sentiment scoring between our system and the annotations in a measurement of mean distance:

$$distance = \frac{1}{|S|}\sum_{p \epsilon S} |r_{ip} - r_{ap}| \tag{5.1}$$

where S represents the set of phrases, p represents each phrase in the set S, r_{ip} represents the rating on phrase p from our sentiment scoring system, and r_{ap} represents the annotated rating on phrase p.

Adverb	Rating	Adverb	Rating
Super	*0.58*	*Pretty*	*0.07*
So	*0.56*	*Too*	*0.05*
Extremely	*0.54*	*Quite*	*0.04*
Incredibly	*0.49*	*Truly*	*0.01*
Very	*0.44*	*A little*	*-0.65*
Really	*0.39*	*A bit*	*-0.83*
Fairly	*0.13*	*Not*	*-3.10*

Table 5-5. Partial results of strength scores calculated for adverbs based on the experimental review corpus. Higher scores represent stronger sentiment strength.

The kappa agreement (Carletta, 1996) between the two annotation sets is 0.68, indicating high consistency between the annotators. As shown in Table 5-6, the obtained mean distance between the scoring from our approach and that from each annotation set is 0.46 and 0.43, respectively, based on the absolute rating scale from 1 to 5. This shows that the scoring of sentiment from our system is relatively close to human annotation. This is easy to understand as the sentiment score of each adjective/adverb is averaged on the ratings over a large user base. The reliability of these results gives us sufficient confidence to make use of these scores as indications of sentiment values.

	Annotation 1	Annotation 2
Mean distance between sentiment score from our system and that from annotations	*0.46*	*0.43*
Kappa agreement between sentiment polarity from our system and that from annotations	*0.55*	*0.60*

Table 5-6. Evaluation of our sentiment scoring system on two annotation sets. The mean distance of sentiment score between our system and two annotations is 0.46 and 0.43 (on a scale of 1 to 5), respectively, which shows that the estimation from our system is relatively close to human judgment. The Kappa agreement on sentiment polarity is 0.55 and 0.60, respectively, indicating moderate consistency between our system and the annotations.

To examine the prediction accuracy of sentiment polarity, for each annotation set, we pooled the phrases with rating 4-5 into "positive," rating 1-2 into "negative," and rating 3 into "neutral." Then we rounded up the sentiment scores from our system to integers and pooled the scores into three polarity sets ("positive," "negative" and "neutral") in the same way. As shown in Table 5-6, the obtained kappa agreement between the result from our system and that from each annotation set is 0.55 and 0.60 respectively. This shows reasonably high agreement on the polarity of sentiment between our system and human evaluation.

5.4 Phrase Classification

In this section, we will present an evaluation of the proposed approach to phrase classification (Liu et al., 2010a), employing the same restaurant-domain review corpus used in the experiments described in previous sub-sections. For training data, we randomly selected 3,000 phrases extracted from the pros/cons of reviews by the linguistic parsing method (the phrases in pros/cons are considered as well-formatted). To generate a human-judgment-consistent training set, we manually labeled the training samples with "<GOOD>" and "<BAD>" labels, based on whether a phrase contains opinion-relevant information (e.g., "delicious pasta: <GOOD>"; "red wine: <BAD>"). We then randomly selected a subset of 3,000 phrases extracted from free-style review texts as the test set, and labeled the phrases with the same "<GOOD>" and "<BAD>" labels as the ground truth. The kappa agreement between the two sets of annotations is 0.73, indicating substantial consistency.

We employed the three types of features (statistical, sentiment and semantic features) as described in Section 3.3 to train the SVMs and the decision tree models for phrase classification. We extracted the unigrams/bigrams from the phrases as statistical features, and employed the sentiment scores calculated for the phrases as the sentiment features.

To extract context-related semantic features, we collected a large pool of well-formatted menus from an online resource[6], which contains 16,141 restaurant menus. Based on the hierarchical structure of these collected menus, we built up a context-related ontology and extracted a set of semantic features from the ontology, such as whether the topic of a phrase is on category-level (e.g., "entrée," "dessert," "appetizers," "salad'), on course-level (e.g., "roasted pork loin," "spicy halibut and clam roast"), or on ingredient-level (e.g., "beans", "chicken," "mushrooms," "scallop").

To extract topic-categorization semantic features, we selected the most frequent 6 topics in the corpus that represented appropriate dimensions for the restaurant domain ("place," "food," "service," "price," "atmosphere" and "portion") as the initial set, and

[6] http://www.menupages.com

clustered the topics of extracted phrases into different aspect categories with the bigram-based topic clustering method. Phrases not belonging to any category were filtered out.

We used these features to train the SVMs and the decision trees as the classification models. To select the most valuable features for model training, we conducted a set of leaving-one-feature-out, or jack-knifing, experiments for both models. We found that all the statistical, sentiment and semantic features except the adjective unigram probability contribute positively to model learning. From further data analysis we observed that many phrases with popular adjectives have context-unrelated nouns (e.g., "good friends" "nice weather"), which means that, although the adjective unigram probability might be high, the phrase is still context irrelevant and is a negative sample. Thus, adjective unigram probability is not a good indicator for phrase relevance. Using the adjective unigram probability as a learning feature will mislead the system into trusting an adjective that is common but has a poor bigram affinity to the context-relevant noun in the phrase. Therefore, we eliminated this feature for both the SVMs and the decision tree learning.

To evaluate the performance of the classification models, we took a set of intuitively motivated heuristic rules as the baseline. Figure 5-5 gives the pseudo-code of the heuristic rule algorithm, which uses variations of all the features except the unigram probability of adjectives.

The performance of classification by different models is shown in Figure 5-6 and Table 5-7. Although the heuristic rule algorithm is complicated and involves human knowledge, both of the statistical models trained by SVMs and the decision tree algorithms outperform the baseline. The SVM model outperforms the baseline by 10.5%

and 11.9% on the two annotation sets, respectively. The decision tree model outperforms the baseline by 16.4% and 23.2% (average relative improvement of 36%), and it also outperforms the SVM model by 5.9% and 11.3% (average relative improvement of 13%).

```
If (sentiment score of the phrase exists)
  If (sentiment score is within neutral range) label=-1;
  else
      if (phrase appeared in the training data)
          if ((3<frequency of phrase < 100))   label = 1;
          else
            if (frequency of phrase >= 100)   label = -1;
              else    if (topic belongs to ontology)  label = 1;
                        else   label = -1;
       else
            if (topic belongs to ontology)   label = 1;
            else   label = -1;
else
  if (phrase appeared in the training data)
           if ((3<frequency of phrase < 100))
                   if (topic belongs to ontology)  label = 1;
                   else   label = -1;
           else
              if (frequency of phrase >= 100)   label = -1;
              else
                   if (topic belongs to ontology)  label = 1;
                   else   if (frequency of noun > 100) label = 1;
                            else   label = -1;
    else
         if (topic belongs to ontology)  label = 1;
         else     if (frequency of noun > 100)   label = 1;
                    else   label = -1;
```

Figure 5-5. Pseudo-code of the heuristic rule algorithm. The rules were manually defined including variations of the features used for SVM and decision tree training.

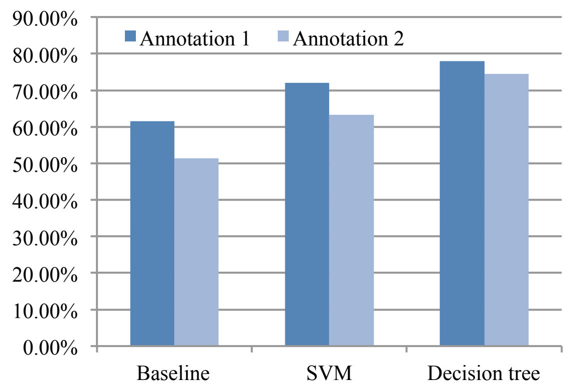

Figure 5-6. Phrase classification performance (measured by precision) on the baseline, the SVM models and the decision tree models on the two sets of annotations.

	Baseline	**SVM**	**Decision tree**
Annotation 1	61.5%	72.0%	**77.9%**
Annotation 2	51.3%	63.2%	**74.5%**

Table 5-7. Precision on phrase classification using the heuristic rule baseline (Baseline), the SVM model, and the decision tree algorithm. On both annotations, the decision tree model outperforms SVM models, which outperform the baseline.

The classification model using the decision tree algorithm can achieve a precision of 77.9% and 74.5% compared with the ground truth. These values indicate the results are quite comparable to human judgment, considering that the precision of one annotation set based on the other is 74% (using one annotation set as the reference and the other as the comparison set). This shows that the decision tree model can predict phrase labels as reliably as human judgment.

To gain further insight on the contributions of each feature to the decision tree learning, Table 5-8 and Figure 5-7 give the results of the experiments on leaving each feature out of model training. Without semantic features, the precision is 70.6% and 65.4% on the two annotation sets, respectively, lower by 7.3% and 9.1% than the case of training the model with all the features (77.9% and 74.5%). This shows that the semantic features contribute to the decision tree learning.

The experimental results also show that the feature of bigram probability of the adjective-noun pair contributes significantly to the model learning. Without this feature, the precision drops by 21.3% and 10.6%. This confirms our observation that although a single adjective is not dominant, the pair of the adjective and the noun that co-occurs with it plays an important role in the classification. The sentiment of phrases also plays an important role. Without sentiment features, the precision drops to 63.4% and 66.6%, respectively, on the two annotations, decreasing by 14.5% and 7.9%.

Feature set	Anotation1	Anotation2
with all features	**77.9%**	**74.5%**
without bigram probability of adjective-noun pair	56.6% (-21.3%)	63.9% (-10.6%)
without unigram probability of the phrase	57.6% (-20.3%)	64.3% (-10.2%)
without unigram probability of the noun	59.8% (-18.1%)	67.8% (-6.7%)
without sentiment score of the phrase	63.4% (-14.5%)	66.6% (-7.9%)
without underlying semantic features	70.6% (-7.3%)	65.4% (-9.1%)

Table 5-8. Performance (precision) of the decision tree model by leaving each feature out of model training. The performance drops with each feature left out, indicating each feature contributes to the model training.

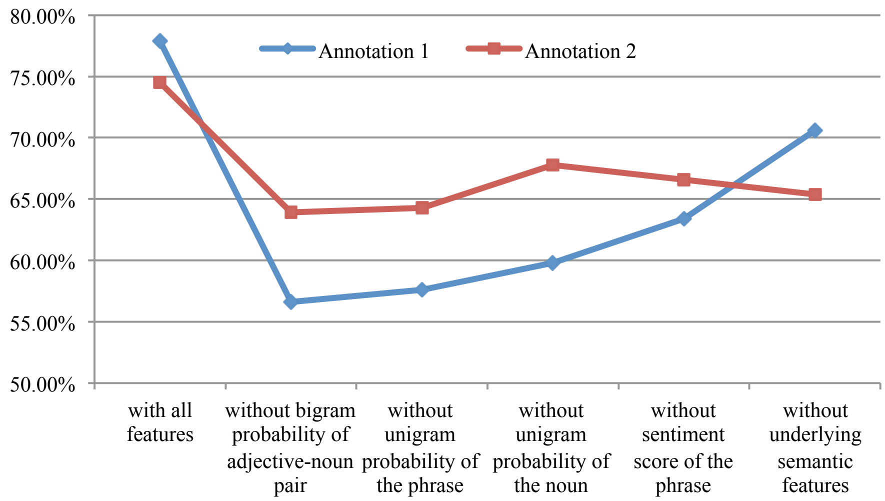

Figure 5-7. Phrase classification performance of the decision tree model by leaving each feature out of model training.

Experimental results show that the decision tree algorithm outperforms the SVMs on this particular classification problem, and it outperforms the heuristic rule baseline significantly. Thus, although the identification of informativeness and relevance of phrases is a rather subjective problem and difficult to predict using only human knowledge, it can be well defined by decision trees. Part of the reason is that the decision tree algorithm can make better use of a combination of Boolean value features (e.g., whether a topic belongs to a context-related ontology) and continuous value features. Also, as the phrase classification task is very subjective, it is very similar to a binary decision problem (Akers, 1978), where decision tree algorithms can fit well.

5.5 Dialogue and Response

To evaluate our proposed framework of developing speech-based interfaces for UGC data access, we applied it to an English restaurant-domain dialogue system. The web-based multimodal spoken dialogue system, *CityBrowser* (Gruenstein and Seneff, 2007), developed in our group, can provide users with information about various landmarks such as the address of a museum, or the opening hours of a restaurant. To evaluate our approaches, we selected the phrases identified as "<GOOD>" by the classification model (as described in Section 5.4) as the candidate pool. These phrases are further catalogued and pruned to create a structured aspect-based summary database. We applied the summary database to the *CityBrowser* system and implemented the database search algorithm as well as the probabilistic language generation within the system (Liu et al., 2010b).

To collect data from real users, we utilized the platform of Amazon Mechanical Turk (AMT)[7] and conducted a series of user studies. To understand what types of queries users may ask, we conducted a first AMT task by collecting restaurant inquiries from general users. Each subject was asked to provide 5 to 20 questions, aiming to find a particular restaurant. We collected 250 sentences through 7 days via the AMT task. We examined the sentence collection and filtered out 40 out-of-domain sentences. From the remaining in-domain sentences, we extracted a set of generic templates encoding the language patterns of all the sentences. We used these templates to automatically create 10,000 sentences for language model training for the speech recognizer. Then we modified the

[7] AMT is an online market place, which gives businesses and developers access to an on-demand scalable workforce. Developers can submit tasks that require human labor. Workers can work at home and choose from thousands of tasks. https://www.mturk.com/mturk/welcome.

context-free grammar to cover these linguistic patterns and created a set of generation rules to extract query key-value pairs. The parsing rate of the modified grammar on the 210 in-domain sentences is 82.4%. We also implemented the database search algorithm for both high-level qualitative questions and feature-specific queries within the dialogue manager (as explained in Chapter 4).

As aforementioned, we have harvested a review corpus from a review publishing web site (*www.citysearch.com*), and collected 857,466 sentences in the restaurant domain, from which we extracted a pool of review phrases. To implement the probabilistic language generation approach, we selected the topics (nouns and noun phrases) in the extracted phrases as the seeds, and used the review sentences in the corpus for linguistic parsing analysis. From the whole corpus, 69 topics were automatically extracted and 2,764 linguistic patterns were identified by our parser (40 patterns for each topic on average). A complete list of topics extracted from the restaurant review corpus and the corresponding seed planting rules can be found in Appendix B. For language generation, we selected the linguistic pattern with the highest probability for each topic.

To evaluate the performance of the system, we conducted another user study on AMT. Our goal is to test whether the system can provide helpful recommendations based on the collected user-published reviews and help users find restaurants of interest. In the AMT task, we presented the system to real users and gave each subject a set of assignments to fulfill. Each assignment is a scenario involving finding a particular restaurant. There are ten HITs (Human Intelligence Tasks) available for each subject. A scenario is randomly assigned in each HIT, and the subject can decide to work on the HIT or skip it. An exemplary scenario is shown in Table 5-9.

"You live in Brighton and you have a friend coming to visit you this weekend. You plan to take him to an American restaurant. You both like some place with nice martinis."

Table 5-9. A scenario example in our user study. Given the scenario, users interact with the system to find the corresponding restaurants via spoken conversations.

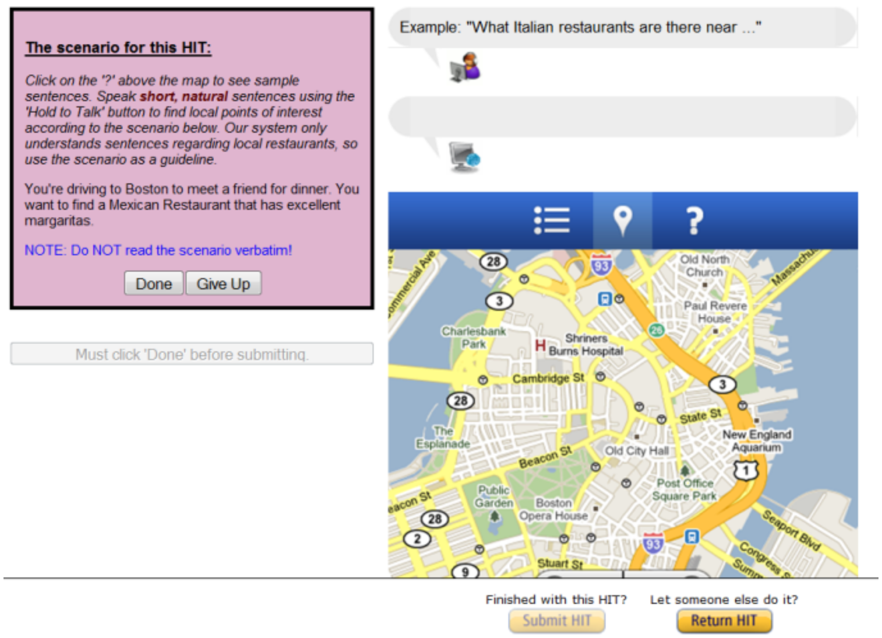

Figure 5-8. Screen shot of our dialogue system in an AMT HIT. On the left-hand side is the scenario defined in this HIT. The right-hand side shows the map for locating the recommended restaurants as well as the conversation history between the user and the system.

To fit in the space of an AMT web page, we utilized a system interface designed to fit the form factor of a handheld mobile device (McGraw et al., 2010). An example of an AMT HIT with our dialogue system is shown in Figure 5-8. The instructions on the left-hand side (titled "The scenario for this HIT") give a randomly selected scenario. The user can talk to the system via a microphone and ask for recommendations for restaurants. The map on the right side will locate the restaurants recommended by the system. The user can click on the "?" button above the map to see suggestion sentences, or click on the bullet mark to see details about the recommended restaurants. When the user is satisfied with the results the system has recommended, he/she can click the button "DONE" and go on to the next HIT.

To obtain a subjective evaluation from general users, we also gave each user a questionnaire and asked them to rate the system on a scale of 1 to 5 for the following questions: "Do you think the system is easy to use?" "Do you find that the recommendations from the system are helpful?" "Do you think the response from the system is in natural language?" Figure 5-9 shows the interface of an AMT HIT after the user has finished the scenario task. The left-hand side shows the questionnaire. The user can choose to answer these questions or skip them. The right side shows the recognition result of the utterance from the user and the response sentences from the system. The recommended restaurant is also shown on the map, with detailed information such as the phone number and the address.

We collected 58 sessions and 34 surveys within 9 days through this AMT task. There are in total 270 utterances collected from the 58 sessions, with an average of 4.6 utterances per session. The length of the utterances varies significantly, from "Thank you" to "Restaurants along Brattle Street in Cambridge with nice cocktails." The average number of words per utterance is 5.3.

We examined the playback of each session. Among all the 58 sessions, 51 of them were successfully fulfilled, i.e., in 87.9% of the cases the system provided helpful recommendations upon the user's request and the user was satisfied with the result.

Among those seven failed cases, one was due to loud background noise, two were due to users' operation errors (e.g., clicking "DONE" before finishing the scenario), and four were due to recognition errors.

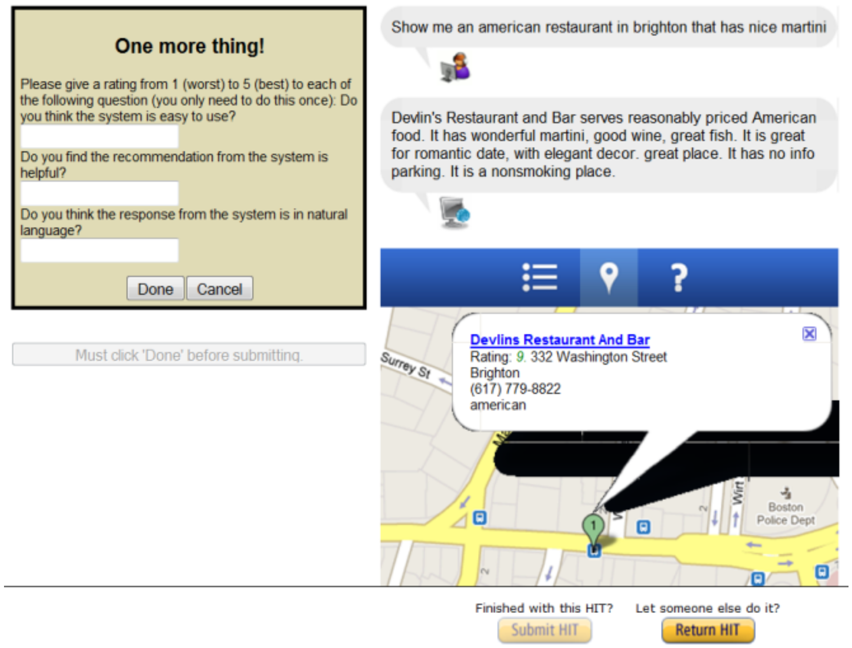

Figure 5-9. Screen shot after a user has finished a HIT. On the left-hand side is a survey for user feedback. On the right-hand side, the map shows the location of the restaurant recommended by the system. The conversation between the user and the system is also shown on the top of the map.

We also examined the feedback from the questionnaire and analyzed the results. On a scale of 1 to 5, the average rating on the easiness of the system is 3.6. The average rating on the helpfulness of the system is 4.4. And the naturalness of the response from the system gets an average rating of 4.1. These numbers indicate that the system is helpful at providing recommendation upon users' inquiries, and the response from the system is presented in a natural way that people could easily understand.

The lower rating of ease of using the system is partially due to recognition errors. For example, a user asked for "pancakes," and the system recommended "pizza places" to him. In some audio clips recorded, the background noise is relatively high. This is unavoidable because many workers on AMT work from home. There were also some utterances that never occurred in our text-based sentence collection for recognizer training, such as "OK, I will take my date to that restaurant tonight," which our system had not yet been trained to handle.

We also allow people to type in textual feedback, which is very helpful for us to tune the system. The AMT experimental platform is very helpful and efficient: it helps us set up real user studies very quickly via the open workspace; it helps us collect and organize user data through a customized API; and it allows us to improve our system incrementally based on users' feedback.

5.6 Chapter Summary

In summary, in this chapter we described multiple experiments on evaluating the proposed approaches with real data. First, we evaluated the lexicon modeling approach with user-annotated data in three domains: restaurant, movie and hotel. We implemented the lexicon expansion and weighting approaches within a semantic tagging system, and evaluated the contribution of both the generative and the discriminative models to slot-level semantic tagging on natural language and keyword-formed queries. We used the query logs from Bing search engine over a year as the external resources for new lexicon learning and weight estimation. Experimental results showed that the expanded lexicons with weights representing the popularity and the relevance of lexicon elements can improve the performance of semantic tagging on both natural language and keyword queries.

We then built a restaurant-domain UGC summary database by collecting a corpus of over 137k user-published restaurant reviews from the Web. From this review corpus, we extracted opinion phrases with the proposed parse-and-paraphrase paradigm, estimated the sentiment value for each phrase, and generated an opinion summary database by selecting the most representative context-relevant phrases.

Experimental results showed that the proposed linguistic parsing techniques can achieve phrase extraction performance over standard keyword matching methods by incorporating high-level linguistic features such as hierarchical semantic structure. The sentiment degree assessment mechanism has also proved to be reliable and close to human judgment, given the fact that the sentiment rating aggregation came from the voting by a large user base.

We also applied the opinion summary database to a multimodal spoken dialogue system. The enhanced restaurant guide system knows about over 100,000 reviews on over 20,000 restaurants in major cities in the States, and can provide users with both general qualitative and feature-specific information about these restaurants via natural spoken conversations.

To evaluate the prototype system, we put the dialogue system on Amazon Mechanical Turk and ask real users to interact with the system. The analysis on the collected spoken utterances and the feedback from users suggested that the system can provide succinct information and make reliable recommendations via natural responses. Such data sharing services via sentiment-involved conversations can greatly increase the capability of spoken dialogue systems and provide active assistance to general users.

Chapter 6

Portability

In previous chapters, we have explored approaches to harvesting and processing unstructured user-generated content from the Web and implementing a speech-based interface for information inquiry. The approaches have been applied mainly in the restaurant domain, but they can be applied to other domains as well, such as movie, hotel, and various product domains.

In this chapter, we will explore the portability of the proposed framework to other domains and other languages. To maximize the divergence from the previously explored restaurant domain, we will investigate a totally different domain -- medical study and health care. And for language portability, we will explore the application to a tone and character-based language -- Mandarin Chinese. We will evaluate the portability of the proposed framework by measuring the effort required for applying it to another domain or another language.

6.1 Domain Portability

Recently, health care and prescription drugs represent a growing topic of user-published discussion online. This is not surprising given that almost half of all Americans take prescription drugs each month, costing over $200 billion in a single year (Gu et al., 2010).

There are various online forums such as *AskAPatient.com* and *WebDB.com* that allow users to post comments and to share with other patients their experiences in using a certain drug. These drug reviews often provide evidence of unusual side effects. By harvesting and analyzing these reviews, we may be able to corroborate suspected side effects or even uncover new knowledge.

U: What side effects do Statin drugs cause?

S: Some of the side effects that show up frequently on Statin drugs are memory problems, difficulty walking, liver damage and all body aches.

U: Can you show me some reviews on memory problems?

S: Some of the drugs in reviews that mention memory problems are Mevacor, Pravachol, Crestor, Lovastatin and Zocor. I've sampled a few reviews relevant to memory problems. *(Top-relevant reviews are displayed on the screen.)*

U: What side effects may co-occur with memory problems?

S: Some side effects that often co-occur with memory problems are kidney damage, muscle problems and Parkinson's disease.

U: What antidepressant drugs can make me gain weight?

S: Some of the antidepressant drugs that mention weight gain are Lexapro, Arnitriptryline hcl, Serzone, Anafranil and Tofrannil-pm. Other side effects in this category are weight loss, inability to lose weight and weight fluctuation. *(Reviews that are most relevant to weight gain are shown on the screen as well.)*

Figure 6-1. A conversation between a user and a drug side effect inquiry system backed by an aggregated database of patient-provided drug reviews ("U" represents the user, and "S" represents the system).

Thus, in this section, we will explore methods to develop speech-based multimodal interfaces for handling spoken/textual queries on prescription-drugs and the associated common side effects backed by a patient-provided drug review database. An example conversation between a user and a spoken dialogue system for inquiring drug-related information is shown in Figure 6-1. In this section, we will investigate how to identify domain-relevant information such as significant side effects from user-generated drug reviews (Liu et al., 2011b), and how to port the proposed speech-interface framework to the new domain.

6.1.1 Data Collection

As a start, we collected over 107K patient-provided reviews on drugs to treat a broad range of problems such as depression, acid reflux disease, high blood pressure, diabetes, etc., from three drug discussion forums ("*AskAPatient.com*," "*Medications.com*" and "*WebDB.com*"). In these forums, patients fill in values for several fields, including the drug name, their age, side effects experienced, and a general comment field where they typically describe their personal story. An example review from *AskAPatient.com* is shown in Figure 6-2.

Here, the field of "Side effects" is similar to the "pros/cons" of restaurant or product reviews, and the field of "Comment" is similar to the free-style review text in the other domains. Although the data format is similar, the task is a little different. The context-relevant information in the restaurant or product domains is normally about the quality of a restaurant/product, mostly containing sentiment-involved opinions. While in the drug domain, the information that people try to obtain by reading reviews from other patients is mostly the effectiveness or the side effects of a certain drug. Different from opinion-involved reviews, drug reviews often have sentiment-free statements, (e.g., "*my legs started to feel heavy after using this drug for three months.*") Such reviews do not necessarily provide explicit sentiment. However, important messages are often provided by context-relevant keyword phrases such as side effect expressions (e.g., "*heavy legs*").

While the generally used context-free grammar for opinion phrases extraction can apply well to sentiment-involved domains such as restaurant and product domains, it does not apply to drug reviews.

Drug: *"Lipitor"*

Dosage: *"20mg 1X D"*

Sex: *"Female"*

Age: *"56"*

Duration: *"5 years"*

Condition: *"Heart disease"*

Side effects: *"Severe muscle pain in shoulders radiating through the chest, cramping in back muscles, calves and hamstrings. Severe muscle pain after working out with weights, all the while losing strength. Difficulty with memory at times..."*

Comment: *"My shoulder pain resulted in a visit to a specialist who said inflammation was present but no torn rotator cuff. Prescribed physical therapy which made it hurt even more. I first noticed the pain several months in to taking the drug. After an ER visit due to severe back spasm/cramp…"*

Figure 6-2. An example drug review from AskAPatient.com. The review contains a few fields, such as the sex and age of the patient, the dosage and duration of drug use, as well as a list of side effects and a free-style comment.

Fortunately, the “Side effects” field of drug reviews provides a very rich set of drug side effects in short phrases, which is a perfect resource for collecting side effect expressions. Thus, to extract the context-relevant information, we first automatically extract from the “*Side effects*” field of each review the words and phrases that describe common side effects on various drugs. We used *n*-grams to obtain a clean set of side effect expressions, including over 7,500 words and phrases, from the corpus of 107K drug reviews (Liu et al., 2011b). We may investigate linguistic features for parsing drug reviews in further work, but here we just use the information that is already available provided by general users, and the focus will be evaluating the effort on extending the speech-based UGC access framework to the new domain.

6.1.2 System Implementation and Evaluation

Our goal is to build a speech-based multimodal interface that allows users to inquire about drug side effects as well as access the patient-provided online reviews. We will evaluate the effort of applying the dialogue system framework to the drug review domain, including speech recognition and synthesis, language understanding, dialogue modeling and response generation, and multimodal interface implementation.

Speech Recognition and Synthesis:

The speech processing components are built in the same way as those for the restaurant domain system. For the speech recognition of users' utterances, we use the SUMMIT system (Glass, 2003). The class *n*-gram language model is trained by parsing a synthetic corpus. To create this corpus, we first created a set of templates of English utterances covering plausible patterns of users' queries, based on the developers' experience and judgment. The templates use a recursive context-free grammar formalism to support a rich set of phrase and sentence patterns. 10,000 utterances were then automatically generated from these templates and were used to train the language model of the recognizer. The templates were later expanded based on real users' speech input from a pilot data collection effort (Liu and Seneff, 2011). For speech synthesis of the system response, we utilize the English text-to-speech system provided by Nokia, the same as the restaurant domain system.

Language Understanding:

Given the user's query utterance generated by the speech recognizer, the system uses a generic syntax-based grammar to parse the utterance (Seneff, 1992a). A few dozens of new domain-relevant rules are added to the generic English grammar to handle domain-specific sentence patterns. The major effort is to enrich the vocabulary to cover domain-relevant words, including 73 drug names (e.g., "mevacor," "lipitor"), 218 domain-relevant nouns and noun phrases (e.g., "wheelchair patients," "painkillers"), and 3,454 side effect expressions (e.g., "ruptured achilles tendon," "osteoarthritis"). These domain-specific words and phrases can be automatically extracted from the drug review corpus (e.g., from the fields of "*Drug name*," "*Condition*" and "*Side effects*" of reviews as shown in Figure 6-2).

The grammar probability model was trained automatically on the corpus of simulated sentences generated from our templates. Similar to the restaurant domain, the parser provides a linguistic frame to encode the syntactic and semantic information of a sentence. A set of generation rules (Baptist and Seneff, 2000) is heuristically constructed to paraphrase the linguistic frame into a set of [key:value] pairs, which represents the semantic meaning of the sentence. Table 6-1 shows some examples of input sentences and the corresponding meaning representation.

Thus, the only difference from the construction of the restaurant-domain system so far is the editing of the templates (for recognizer/grammar training) that are aimed to cover possible patterns of users' input utterances, and the context-free grammar as well as the generation rules used to handle the semantic meaning representation for domain-specific information.

Sentence I	*What antidepressant drugs can make me gain weight?*
[Key:value] pairs	Drug class: antidepressant; drug name: *what*; side effect: weight gain; category: weight problems
Sentence II	*Does Lipitor cause headache or general weakness?*
[Key:value] pairs	Drug class: Statins; drug name: Lipitor; side effect #1: headache; category #1: cognition problems; side effect #2: general weakness; category #2: muscle problems
Sentence III	*What side effects often co-occur with heart failure when using Statin drugs?*
[Key:value] pairs	Side effect: heart failure; category: heart problems; command: list_co_occur_side_effects; drug class: Statins
Sentence IV	*Can you show me the reviews on SSRI related to memory loss?*
[Key:value] pairs	Drug class: antidepressant; drug group: SSRI; side effect: memory loss; category: cognition problems; command: list_reviews

Table 6-1. Examples of [key:value] pairs generated from users' utterances of various queries. Sentence I queries about drug names. Sentence II asks about particular side effects. Sentence III inquires about co-occurrence relations between multiple side effects. And Sentence IV asks for original reviews.

Dialogue Modeling and Response Generation:

To handle domain-specific queries, we added some new commands to the database search algorithms. For example, in Sentence I in Table 6-1, "*drug name: *what**" will trigger a database search for drug names ("**what**" indicates a search on this key). Thus, the system will retrieve as candidates those drug names which are strongly associated with this specific side effect ("weight gain") from the drug review database.

Another example is Sentence IV in Table 6-1. When users ask the system to show some reviews about a side effect related to a specific drug or drug group, a review searching event will be triggered by the command "*list_reviews*". Reviews on this particular drug are ranked by their relevance to the specific side effect using standard ranking algorithms (Robertson et al., 1994) and the top-ranked reviews are retrieved and displayed on the GUI (graphical user interface).

Multimodal Interface Implementation:

We use the open source WAMI[8] toolkit (Gruenstein et al., 2008) to embed the recognizer, the synthesizer, the language understanding and generation components, and the dialogue manager into a web-based interface, similar to the restaurant-domain system. With this portable toolkit, it requires little effort to create the web-based interface for the extended domain.

Figure 6-3 shows the GUI of the drug-domain spoken dialogue system (Liu and Seneff, 2011). Users can talk to the system by clicking the microphone icon (on the top right of the interface). The conversation history is shown in text on the top, and users can browse previous dialogue turns. Below the history window, there is a type-in window where users could type their questions in text instead of speaking. For each review displayed in the middle of the interface, the side effect phrases that were extracted from the review are listed, to serve as a succinct summary (e.g., "*keywords: general pain, depression, aggressive behavior, memory problem*s" for Review #4 in Figure 6-3), and the specific

[8] WAMI is an open-source toolkit developed in our group to add speech recognition capabilities to a web page for developing, deploying and evaluating Web-accessible multimodal interfaces. http://wami.csail.mit.edu/

side effect inquired by the user is highlighted in red. Users can browse through the displayed set of review summaries and open up any summary to see the expanded text.

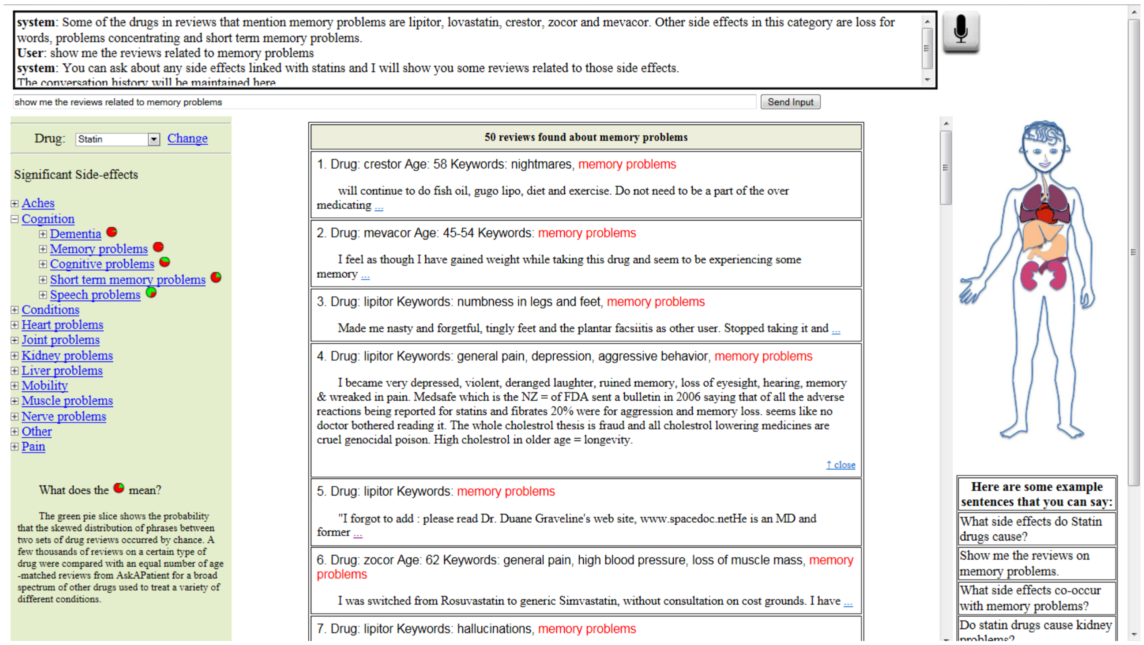

Figure 6-3. Screenshot of the speech-based multimodal interface for drug side effect inquiring and review browsing. Dialogue history is shown on the top of the review browsing window. On the left of the window is a hierarchical tree of side effects catalogued into different condition types.

Experiments and Evaluation:

For the system evaluation, we conducted two runs of data collection (Liu and Seneff, 2011). In each user session, we first show the user a demo video to demonstrate the capabilities of the system. Then the user is asked to conduct a short practice interaction, in order to become comfortable with our speech interface and the overall system configuration. When the user is ready, he/she starts a new session as the formal recording session. Users' utterances were captured through a headset microphone and later transcribed.

From the initial data collection episode, we collected 198 utterances from real users. This set of utterances is used as the training set to improve the grammar coverage and speech recognition performance. A total of 184 utterances were collected from the second data collection episode, which were taken as the test set. The parsing rate on the training and test sets was 98.0% and 94.0% respectively, and the word error rate in speech recognition was 11.8% and 27.3% (Liu and Seneff, 2011). A questionnaire was also collected for a subjective evaluation, where each user was asked to give a numerical rating on a Likert scale of 1 to 5 for each of the following questions: 1) Did the system understand your questions well? 2) Is the information from the system helpful? 3) Were the responses from the system natural? 4) Is the interface to the system intuitive and easy to use? 5) Does the speech interface enhance the system? Across all the subjects, the average perceived rating on each question is 3.0, 3.3, 3.7, 4.0 and 4.3 respectively.

6.2 Language Portability

In this section, we will describe our continuing efforts to explore the portability issue regarding to language. Specifically, we will derive a Mandarin dialogue system, CityBrower II (Liu et al., 2008), from its English predecessor – a restaurant-domain dialogue system. We will focus on the speech interface implementation regarding to language difference, without spending efforts on handling Chinese restaurant reviews. An example of a conversation between a user and the CityBrowser II system is shown in Figure 6-4.

U: 我想找一家日本料理餐馆.

S: 北京有 200 多家日本料理餐馆，它们大多集中在海淀区和朝阳区.

U: 海淀区知春路上有吗?

S: 知春路上有四家日本料理餐馆，它们是江织田日本料理，和岛屋日本料理，元绿回转寿司，和一番日本料理。

U: 哪一个比较便宜一点儿.

S: 一番日本料理比较便宜，它位于知春路 49 号地下一层.

U: 订餐电话是多少?

S: 电话号码是 88099658.

U: 清华附近呢?

S: 清华大学附近有 5 家比较便宜的日本料理餐馆，它们是千浔日本料
料
理，唐宫海鲜舫，银座餐厅，红叶日本料理，和千鹤料理屋。

U: 有川菜吗?

S: 清华大学附近有 14 家川菜餐厅，它们大多数都比较便宜.

Figure 6-4. Example of a conversation between a user and the CityBrowser II system in Mandarin Chinese. “U” represents the user, and “S” represents the system.

6.2.1 Database Construction

Our first step in developing CityBrowser II is to replace the original English restaurant database with a Mandarin one. The data are gleaned from crawling restaurant databases available on the web[9], which contains information about 6,533 restaurants in Beijing, China. Restaurants are organized geographically around neighborhoods of 66 major universities.

[9] http://www.eatwell.com.cn/

In spoken language Chinese people tend to refer to a named entity with a much shorter nickname, e.g., “全聚德” for “北京全聚德烤□店.” In English, it is easier to filter the database with partial query match (e.g., “Starbucks” for “Starbucks coffee”), since English words are separated by space. Chinese words, however, are not separated by spaces, and thus require explicit tokenization. Thus, we generate a list of nicknames for each named entity with a few heuristic rules. The generalization procedure works as follows: first, we optionally remove all aliases of the word “restaurant” from restaurant names, which include a set of more than 40 alternative forms in Chinese, such as “酒楼” “酒店” “□庄” “酒家.” Then we remove locative nouns from the beginning of the named entity, such as “北京.” Lastly, we remove the description of the cuisine, such as “烤□” and “火□.” The named entities can then be referred to by these derived variants on different levels (e.g., “全聚德,” “全聚德烤□” or “北京全聚德”).

We have also implemented a procedure to handle the restaurant hours, which requires more sophistication as there are various ways to express time intervals in Mandarin. We regularize different expressions into a uniform format and convert values in this format into Mandarin, such as from “10:30–1:40 6:00--0:00” to “上午十点半到下午一点四十, 晚上六点到凌晨十二点”. An example of a cleaned-up database entry is shown in Table 6-2. We keep the names of the information fields the same as in the English system (e.g., “name,” “street,” “phone,” “cuisine,” “hours” and “price”), such that the database search algorithms used in the English system can be inherited.

Name	"北京全聚德烤鸭店"
Nickname	"全聚德烤鸭" "北京全聚德" "全聚德"
Branch	"北图分店"
Address	"海淀区白石桥路 37 号"
District	"海淀"
Street	"白石桥路"
Street number	"37"
Phone	"68420527"
Cuisine	"烤鸭"
Neighbourhood	"民族大学" "中央民族大学" "舞蹈学院" "北京舞蹈学院"
Hours	"上午十点半到下午一点半，下午四点半到晚上九点半"

Table 6-2. An example of a database entry in the Mandarin Chinese restaurant-domain system. The fields of information are maintained in English, and the values of contents are in Chinese characters.

6.2.2 System Implementation and Evaluation

In this section, we will describe how to transform a pre-existing dialogue system from English to Chinese, focusing on speech recognition and synthesis, language understanding, dialogue modeling and response generation, and multimodal interface implementation.

Speech Recognition and Synthesis:

The speech processing components are built in a similar way to the English system. The acoustic models for the speech recognition system, SUMMIT (Glass, 2003), are trained with a combination of two Chinese data corpora, “Yinhe” (Wang et al., 1997) and “MAT2000” (Wang et al., 2000), both of which contain Mandarin Chinese speech data from native speakers. The class *n*-gram language model is trained by parsing a corpus (Seneff, 1992a). We make use of the corpus from the English system and create templates from translations into Mandarin of the English utterances. Then we automatically generate over 11,000 utterances in Mandarin from these translated templates (Liu et al., 2008). For the Mandarin synthesizer, we utilize a Mandarin text-to-speech system provided by the Industrial Technology Research Institute (ITRI).

Language Understanding:

To convert language understanding from English to Mandarin Chinese, we supply a generic-domain Mandarin grammar (Xu et al., 2008) with a set of restaurant-domain class specifications. For example, we specify a Chinese cuisine category as an adjectival class covering 104 cuisines in Mandarin (e.g., “川菜,” “粤菜,” “火锅”). We also extend a Mandarin vocabulary of the generic domain with 500 domain-specific entries, mostly Chinese restaurant names and street names.

Input utterance	*<我想去一家在海淀区知春路上便宜一点儿的川菜餐馆>*	
Key-value pairs	**Unchanged**	**Changed**
	clause: request topic: restaurant price_range: cheap	cuisine: 川菜 city: 海淀 street: 知春路

Table 6-3. Example of key-value pairs generated from a Mandarin Chinese utterance. The keys and generic values such as “clause,” “topic” and “price_range” are the same as in the English system. Only database contents are represented with Chinese characters (e.g., “cuisine,” “city” and “street”).

To keep the semantic representation language-independent, we maintain the keys and generic values of [key:value] pairs in English, while representing the values of database contents directly with Chinese characters, as exemplified in Table 6-3. In this way, the meaning representation framework as well as the database search algorithms can be inherited from the English system, while the database content can be filtered with inquired Chinese keywords.

Dialogue Modeling and Response Generation:

The dialogue manager is language-independent and can be inherited from the English system unchanged. We use the same language generation system, GENESIS (Baptist and Seneff, 2000), for response generation, which creates a well-formed Mandarin string from a response frame via generation rules. We can inherit the set of generation rules used in the English system. A main issue of translating generation templates from English to Mandarin Chinese is word order differences. This requires expert knowledge and can be fixed with manual effort.

Multimodal Interface Implementation:

Like its predecessor, CityBrowser II is a web-based conversational system and is implemented with the WAMI (Gruenstein et al., 2008) toolkit. A screenshot of the system appears in Figure 6-5, which consists of a dynamic web page centered on a Google map implemented with Google Ditu API[10], a Chinese map API. Audio communication is controlled via a Java applet (the green button on the top of the map) embedded in the page which provides a push-to-talk button and endpointing. The transcribed user's

[10] http://ditu.google.cn/

utterance is shown below the audio applet. A list of currently in-focus entities are shown on the map with balloon icons, as well as displayed to the right of the map. The GUI (graphical user interface) implementation can be directly inherited from the English system. The only change is the use of Chinese geocoding API instead of the English version.

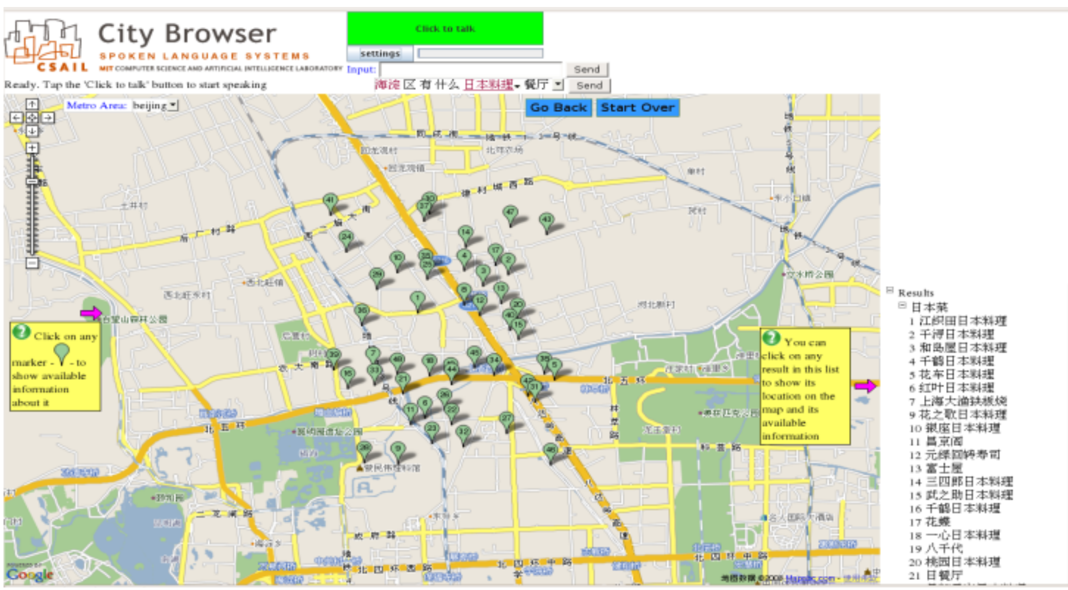

Figure 6-5. Screenshot of geographic GUI of CityBrowser II. Users can talk to the system by clicking the Java applet on the top of the map. Retrieved entities are shown on the map and listed on the right-hand side. Users could also type in their utterances in the type-in window below the audio applet.

Experiments and Evaluation:

We performed a preliminary system evaluation by logging the interactions of 10 subjects with the system (Liu et al., 2008). All of the subjects are native speaker of Mandarin Chinese. Each subject went through a supervised practice session, where they were taught how to record and led through 15 single-turn interactions and three scenarios. Each subject was then given 10 scenario-based tasks with gradually increasing complexity. An example scenario is shown in Table 6-4.

You are a student at Tsinghua University. You have a friend visiting you, who likes Italian food very much. Your plan of the day is to first take him for lunch at a Japanese restaurant near school, then go window shopping at Wangfujing Street. After shopping in the afternoon, you will treat him at a fancy Italian restaurant nearby. You need to make a reservation at the restaurant by phone, and you want to know exactly the location of the restaurant.

Table 6-4. An example scenario-based task. Given the scenario, users interact with the system via spoken conversations to find the corresponding restaurant defined in the scenario.

A total of 836 recorded utterances were collected from these subjects. Eight of the 10 subjects succeeded in all 10 tasks, and the other two subjects succeeded in 9 out of 10 tasks. In addition to the interactions with the system, we also conducted a post-test survey on each subject. Each subject was given a survey to see if he/she finds the system easy to use, whether he/she thinks the restaurant-guide helpful, and whether he/she would like to recommend the system to their friends. Each question was evaluated on a Likert scale of 1 to 5 (1: very difficult, 5: very easy; 1: not helpful, 5: very helpful; 1: no recommendation, 5: highly recommend). Across all the subjects, the average perceived ease of use was 3.8; the average perceived helpfulness was 4.6; and the average recommendation was 4.0.

6.3 Chapter Summary

In summary, to explore the portability of the proposed framework, we described a spoken dialogue system implemented in an extended domain, the prescription-drug domain. The Web-based multimodal system allows consumers of prescription drugs to discover possible side effects revealed from patient-provided drug reviews and to access relevant reviews from health discussion sites efficiently via speech, text and gestures.

The construction of the spoken dialogue system inherited most of the components of the previously explored restaurant-domain system, and made only changes to the speech recognizer training (template editing for language model training), the context-free grammar expansion (for handling domain-specific information), and the dialogue modeling capacity expansion (customizing new commands for domain-specific queries).

For speech recognition, the only difference from the construction of the restaurant-domain system is the editing of the templates (for recognizer/grammar training) that are aimed to cover possible patterns of users' input utterances. For language understanding, the context-free grammar and the generation rules are enriched with domain-specific information (e.g., drug-related vocabularies) which can be discovered from patient-generated reviews automatically. In total, around 4,000 domain-specific words were automatically detected from the corpus and added to the generic English grammar, and around 260 generation rules were added for the meaning extraction for these domain-specific words.

For dialogue modeling, the database search algorithms can be inherited from the system of the original domain, and a few customized commands can be added according to the specific needs of the new domain. The web-based graphical interface is implemented with the portable WAMI toolkit and requires little extra effort.

To evaluate the language portability, we transferred the English restaurant-domain system into a totally different language, Mandarin Chinese, focusing on the speech interface implementation regarding to language difference. The system inherited most of the components of the predecessor English system and required only relatively minor changes to account for the differences between English and Mandarin Chinese. The major effort required is the translation of templates used to generate sentences for recognizer training, as well as the translation of language generation templates from the original language to the target one. The context-free grammar needs to be switched to the target language, but the meaning representation can be maintained in the original language, with only the values of database contents represented in the target language.

We make use of the sentence corpus from the English system and create templates from translations into Mandarin of the English utterances. These translated templates can be used to generate sentences for training language models used by the speech recognizer. The acoustic models are trained on Mandarin Chinese speech data. For language understanding, we supply a generic-domain Mandarin grammar with a set of domain-specific class specifications.

The database search algorithms and dialogue modeling strategies can be inherited from the predecessor of English system without any change. We can inherit the language generation rules used in the English system as well. The only challenges are word order difference when translating the English templates into Mandarin Chinese, which requires expert knowledge and can be handled manually. The WAMI toolkit is used for interface implementation, thus no effort is required expect the change of Google map API from English to Chinese.

This shows that the proposed framework is reasonably portable. With small efforts on handling domain-specific or language-specific information, it can be applied to other domains or other languages efficiently and effectively.

Chapter 7

Summary and Future Work

In this thesis, we have explored a framework for developing a speech-based interface for user-generated content access. The challenges are mainly on harvesting and processing unstructured UGC data, and developing multimodal interfaces for information access via speech, text and gestures. In this chapter, we will summarize the work we have done on exploring possible approaches to tackle these challenges, and we will discuss the future work.

7.1 Summary

The Web has been exploding dramatically with user-generated-content (UGC) over the past decade, such as social networks (e.g., Facebook, Twitter and YouTube) and community-contributed sites (e.g., WikiPedia, Yelp and TripAdvisor). At the same time, there is a fast-increasing usage in mobile devices such as smart phones and tablets along with the rapid development of application software (Apps), especially for visiting social networks and online communities. However, the user-generated data available on the Web are growing exponentially and it is impossible for people to digest all the information even with the help of various applications on mobile devices.

To help users obtain information more efficiently, both the interface of content access and the representation of information need to be improved. An intuitive and personalized interface could be a human-like assistant that can engage a user in a continuous spoken dialogue to capture the user's intent proactively, and summarize the information on the Web in a concise manner as well as presenting it to the user in a natural way.

This thesis explained our research on a general framework for developing such conversational systems which can process user-generated content and present the summarized information with natural dialogue interaction. In Chapter 1, we reviewed the state of the art in the areas of user intention understanding, spoken dialogue systems and unstructured data summarization. We investigated the remaining challenges in these areas and defined the scope of our research.

To better understand users' intentions from users' spoken/textual input, in Chapter 2 we proposed a lexicon modeling approach to contextual query interpretation via CRF-based semantic tagging. External resources such as user-query logs were employed for automatic lexicon expansion. Both generative models and discriminative models were explored for lexicon weighting. Experiments on various domains showed that the lexicon modeling approach can expand pre-existing lexicons automatically, and the expanded lexicons as well as the lexicon weights learned from users' click statistics can improve semantic tagging performance effectively.

To summarize large amount of user-generated data into a condensed and structured database, we presented in Chapter 3 a framework for preprocessing unstructured UGC data. We proposed a parse-and-paraphrase approach to extracting representative phrases from unstructured sentences, as well as introducing an algorithm for assessing the degree of sentiment in opinion expressions based-on user-generated ratings. We also used a phrase classification model employing semantic, statistical and sentiment features to select context-relevant phrases automatically for creating succinct, descriptive and catalogued UGC summaries as well as aspect ratings.

To generate natural spoken responses automatically, in Chapter 4 we proposed a probabilistic predicate selection approach that learns linguistic patterns from user-generated corpus. The best-match predicates for various topics can be learned automatically through the probability statistics from the user-generated corpus. A generic dialogue management framework was also introduced, which supports the generation of opinion-sharing conversations based on aggregated UGC database.

To evaluate the framework, we collected a user-generated review corpus in the restaurant domain from the Web, which was processed through the proposed pipeline of unstructured data processing. The experiments were described in Chapter 5. A restaurant-domain recommendation system enhanced by this framework was implemented as a demonstration. Users can interact with the system via speech or typed-in text to inquire about restaurants and ask for recommendations on various dimensions such as service, food or ambiance. The interactions between real users and the prototype system were monitored for system evaluation.

To demonstrate the portability of the approaches, in Chapter 6 we described our efforts on applying the proposed framework in a totally different domain as well as in another language. Based on the framework, we implemented a dialogue system in the health care domain, which provides users with a rich facility for exploring the association of prescription drugs with possible side effects and browsing through patient-provided drug reviews via speech, text and gestures. We also transferred the English restaurant-domain system into a Mandarin Chinese system, inheriting the structure of the predecessor English system with minor changes to account for language difference.

7.2 Future Work

In this thesis, we have explored a universal framework that supports multimodal access to user-generated content, with a speech-navigated web-based interface and a generalized platform for unstructured data processing. The contribution of this work lies in that it advances the integration of unstructured data summarization and speech-based human-computer interaction. With the help of such dialogue systems, users can access the online community-edited information more effectively and more efficiently.

For future work, we plan to collect more speech data from real users. We will maintain online versions of the systems we implemented in various domains and languages, and make them available to the general public. We will continue to improve the performance of these systems through larger-scale data collections from general users.

Another direction is to develop a speech interface for harvesting spoken UGC data (e.g., spoken reviews, comments), which can allow users to add their own experience on restaurants, movies, drugs, etc., through natural speech and text. We will also explore crowd-sourcing methods to aid in the transcription of these recordings, such as relying on general users via Amazon Mechanical Turk.

A more ambitious future goal is to develop a speech-based platform for general domains. Currently, dialogue systems in different domains are implemented independently. A more practical application would be an integrated system, which can interact with users in continuous conversations across multiple domains. A domain-transparent speech-navigated platform that can provide users with various knowledge and easy access to online data would greatly benefit people's daily lives in the future.

Bibliography

[1] Sheldon B. Akers. 1978. Binary Decision Diagrams. *IEEE Transactions on Computers*, C-27(6):509–516.

[2] James Allen, Myroslava Dzikovska, Mehdi Manshadi, and Mary Swif. 2005. Deep Linguistic Processing for Spoken Dialogue Systems. *In Proceedings of the 5th Workshop on Important Unresolved Matters*, pages 49–56.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Multi-facet Rating of Product Reviews. *In Proceedings of European Conference on Information Retrieval.*

[4] Rajesh Balchandran, Leonid Rachevsky, Larry Sansone, and Roberto Sicconi. 2009. Dialog System for Mixed Initiative One-Turn Address Entry and Error Recovery. *In Proceedings of SIGDIAL meeting on Discourse and Dialogue.*

[5] Lauren Baptist and Stephanie Seneff. 2000. Genesis-II: A Versatile System for Language Generation in Conversational System Applications. *In Proceedings of International Conference on Spoken Language Processing (ICSLP)*.

[6] Frédéric Béchet. 2008. Processing Spontaneous Speech in Deployed Spoken Language Understanding Systems: A Survey. *In proceedings of ACL/IEEE Workshop on Spoken Language Technology*.

[7] Jonas Beskow, Jens Edlund, Björn Granström, Joakim Gustafson, Gabriel Skantze, and Helena Tobiasson. 2009. The MonAMI Reminder: A Spoken Dialogue System for Face-to-Face Interaction. *In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech).*

[8] Dan Bohus and Alexander Rudnicky. 2003. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. *In Proceedings of European Conference on Speech Communication and Technology (Eurospeech)*.

[9] H. Bonneau-Maynard, S. Rosset, C. Ayache, A. Kuhn, and D. Mostefa. 2005. Semantic Annotation of the French Media Dialog Corpus. *In Proceedings of In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*.

[10] S.R.K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. 2008. Learning Document-Level Semantic Properties from Free-Text Annotations. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[11] Stephan Busemann and Helmut Horacek. 1998. A Flexible Shallow Approach to Text Generation. *In Proceedings of the Ninth International Workshop on Natural Language Generation.*

[12] Giuseppe Carenini, Raymond Ng, and Adam Pauls. 2006. Multi-Document Summarization of Evaluative Text. *In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

[13] Jean Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2), pages 249–254.

[14] Pimwadee Chaovalit and Lina Zhou. 2005. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. *In Proceedings of the 38th Hawaii International Conference on System Sciences.*

[15] Hang Cui, Vibhu Mittal, and Mayur Datar. 2006. Comparative Experiments on Sentiment Classification for Online Product Reviews. *In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI).*

[16] Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *In Proceedings of the International Conference on World Wide Web (WWW).*

[17] Stephen Della Pietra, Mark Epstein, Salim Roukos, and Todd Ward. 1998. Fertility Models for Statistical Natural Language Understanding. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 168-173.

[18] Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Re-Ranking Models for Spoken Language Understanding. *In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

[19] John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Doug Moran. 1993. Gemini: A Natural Language System for Spoken-Language Understanding. *In Proceedings of the Thirty-First Annual Meeting of the Association for Computational Linguistics (ACL).*

[20] W. Eckert, T. Kuhn, H. Niemann, S. Rieck, A. Scheuer, and E. G. Schukat-talamazzini. 1993. A Spoken Dialogue System for German Intercity Train Timetable Inquiries. *In Proceedings of European Conference on Speech Communication and Technology (Eurospeech).*

[21] Koji Eguchi and Victor Lavrenko. 2006. Sentiment Retrieval using Generative Models. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 345–354.

[22] Michael Elhadad and Jacques Robin. 1992. Controlling Content Realization with Functional Unification grammars. *Aspects of Automated Natural Language Generation*, pages 89-104.

[23] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. *In Proceedings of the 5th Conference on Language Resources and Evaluation.*

[24] Mark Gales and Steve Young. 2008. The Application of Hidden Markov Models in Speech Recognition. *Foundations and Trends in Signal Processing*, 1(3).

[25] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. Pulse: Mining Customer Opinions from Free Text. *In Proceedings of the 6th International Symposium on Intelligent Data Analysis.*

[26] James Glass. 2003. A Probabilistic Framework for Segment-Based Speech Recognition. *Computer Speech and Language*, *Publisher: Elsevier*, 17(2-3):137-152.

[27] Andrew Goldberg and Xiaojin Zhu. 2006. Seeing Stars when there aren't Many Stars: Graph-based Semi-Supervised Learning for Sentiment Categorization. *In HLT-NAACL Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*.

[28] A. L. Gorin, B. A. Parker, R. M. Sachs, and J. G. Wilpon. 1997. How May I Help You? *Speech Communication*, 23(1-2):113–127.

[29] Alexander Gruenstein and Stephanie Seneff. 2007. Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms. *In Proceedings of the 8th SIGDIAL Workshop on Discourse and Dialogue*, pages 111-119.

[30] Alexander Gruenstein, Ian McGraw, and Ibrahim Badr. 2008. The WAMI Toolkit for Developing, Deploying, and Evaluating Web-Accessible Multimodal Interfaces. *In Proceedings of the Tenth International Conference on Multimodal Interface (ICMI).*

[31] Qiuping Gu, Charles F. Dillon, and Vicki L. Burt. 2010. Prescription Drug Use Continues to Increase: U.S. Prescription Drug Data for 2007-2008. *NCHS Data Brief*, (42):1.

[32] Joakim Gustafson, Linda Bell, Jonas Beskow, Johan Boye, Rolf Carlson, Jens Edlund, Björn Granström, David House, and Mats Wirén. 2000. AdApt a Multimodal Conversational Dialogue System in an Apartment Domain. *In Proceedings of International Conference on Spoken Language Processing (ICSLP)*.

[33] Stefan Hahn, Patrick Lehnen, Christian Raymond, and Hermann Ney. 2008. A Comparison of Various Methods for Concept Tagging for Spoken Language Understanding. *In Proceedings of the Language Resources and Evaluation Conference (LREC).*

[34] Dilek Hakkani-Tr and Gokhan Tur. 2007. Tutorial on Spoken Language Understanding. *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[35] Vasileios Hatzivassiloglou and Janyce Wiebe. 2000. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. *In Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 299-305.

[36] Yulan He and Steve Young. 2005. Spoken Language Understanding Using the Hidden Vector State Model. *Speech Communication*, 48(3-4):262-275.

[37] Renate Henschel and John Bateman. 1997. Application-Driven Automatic Sub-Grammar Extraction. *CoRR cmp-lg/9711010.*

[38] Ryuichiro Higashinaka, Rashmi Prasad, and Marilyn Walker. 2006. Learning to Generate Naturalistic Utterances Using Reviews in Spoken Dialogue Systems. *In Proceedings of the joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL).*

[39] Ryuichiro Higashinaka, Marilyn Walker, and Rashmi Prasad. 2007. An Unsupervised Method for Learning Generation Dictionaries for Spoken Dialogue Systems by Mining User Reviews. *Journal of ACM Transactions on Speech and Language Processing*.

[40] Minqing Hu and Bing Liu. 2004a. Mining and Summarizing Customer Reviews. *In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[41] Minqing Hu and Bing Liu. 2004b. Mining Opinion Features in Customer Reviews. *In Proceedings of Nineteenth National Conference on Artificial Intelligence (AAAI).*

[42] Torsten Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *In Proceedings of 10th European Conference on Machine Learning (ECML)*, pages 137–142.

[43] Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, MarilynWalker, Steve Whittaker, and Preetam Maloor. 2002. MATCH: An Architecture for Multimodal Dialogue Systems. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[44] Soo-Min Kim and Eduard Hovy. 2006. Automatic Identification of Pro and Con Reasons in Online Reviews. *In Proceedings of the joint conference of the*

International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL), pages 483–490.

[45] Lun-Wei Ku, Li-Ying Lee, Tung-Ho Wu, and Hsin-His Chen. 2005. Major Topic Detection and its Application to Opinion Summarization. *In Proceedings of the Annual International ACM SIGIR Conference (SIGIR)*, pages 627-628.

[46] Roland Kuhn and Renato De Mori. 1995. The Application of Semantic Classification Trees to Natural Language Understanding. *IEEE transactions on pattern analysis and machine intelligence*, 17(5):449--460.

[47] Xiao Li. 2010. Understanding the Semantic Structure of Noun Phrase Queries. *In Proceedings of Association for Computational Linguistics.*

[48] Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting Structured Information from User Queries with Semi-Supervised Conditional Random Fields. *In Proceedings of the Annual International ACM SIGIR Conference (SIGIR)*.

[49] Rensis Likert. 1932. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 22(140):1--55.

[50] Bing Liu. 2011. Web Data Mining: Exploring Hyperlinks, *Contents and Usage Data*, *Second Edition*, *Springer*.

[51] Bing Liu, Chee Wee Chin, and Hwee Tou Ng. 2003. Mining Topic-Specific Concepts and Definitions on the Web. *In Proceedings of the twelfth international World Wide Web conference (WWW).*

[52] Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. *In Proceedings of International Conference on World Wide Web (WWW).*

[53] Jingjing Liu, Xiao Li, Alex Acero, and Ye-Yi Wang. 2011a. Lexicon Modeling for Query Understanding. *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[54] Jingjing Liu, Alice Li and Stephanie Seneff. 2011b. Automatic Drug Side Effect Discovery from Online Patient-Submitted Reviews: Focus on Statin Drugs. *In proceedings of the First International Conference on Advances in Information Mining and Management*.

[55] Jingjing Liu and Stephanie Seneff. 2009. Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm. *In proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[56] Jingjing Liu and Stephanie Seneff. 2011. A Dialogue System for Accessing Drug Reviews. *In proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*.

[57] Jingjing Liu, Stephanie Seneff, and Victor Zue. 2010a. Dialogue-Oriented Review Summary Generation for Spoken Dialogue Recommendation Systems. *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

[58] Jingjing Liu, Stephanie Seneff, and Victor Zue. 2010b. Utilizing Review Summarization in a Spoken Recommendation System. *In Proceedings of SIGDIAL meeting on Discourse and Dialogue*.

[59] Jingjing Liu, Yushi Xu, Stephanie Seneff, and Victor Zue. 2008. CityBrowser II: A Multimodal Restaurant Guide in Mandarin, *In Proceedings of the International Symposium on Chinese Spoken Language Processing (ISCSLP)*.

[60] Ian McGraw, Chia-Ying Lee, Lee Hetherington, Stephanie Seneff, and James Glass. 2010. Collecting Voices from the Cloud. *In Proceedings of the Language Resources and Evaluation Conference (LREC).*

[61] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. *In Proceedings of International Conference on World Wide Web (WWW)*.

[62] Mehryar Mohria and Fernando Pereira. 2002. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech & Language*, 16(1):69--88.

[63] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining Product Reputations on the Web. *In Proceedings of the 8th Internatinal Conference on Knowledge Discover and Data Mining*.

[64] Tony Mullen and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. *In proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[65] Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. 2006. Examining the Role of Linguistics Knowledge Sources in the Automatic Identification and Classification of Reviews. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[66] Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic Language Generation for Spoken Dialogue Systems. *In Proceedings of the ANLP/NAACL Workshop on Conversational Systems.*

[67] Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[68] Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[69] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

[70] Marius Pasca and Benjamin Van Durme. 2007. What You Seek Is What You Get: Extraction of Class Attributes from Query Logs. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

[71] Marius Pasca and Benjamin Van Durme. 2008. Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. *In Proceedings of the Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT).*

[72] Ana-Maria Popescu and Oren Etzioni. 2005. Extracting Product Features and Opinions from Reviews. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

[73] J. R. Quinlan, 1986. Induction of Decision Trees. *Machine learning, Springer-Netherlands*.

[74] Lawrence R. Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *In Proceedings of the IEEE*, 77(2):257–286.

[75] Owen Rambow, Srinivas Bangalore, and Marilyn Walker. 2001. Natural Language Generation in Dialog Systems. *In Proceedings of the First International Conference on Human Language Technology Research*.

[76] Antoine Raux, Brian Langner, Alan W Black, and Maxine Eskenazi. 2003. LET'S GO: Improving Spoken Dialog Systems for the Elderly and Non-natives. *In Proceedings of European Conference on Speech Communication and Technology (Eurospeech)*.

[77] Christian Raymond and Giuseppe Riccardi. 2007. Generative and Discriminative Algorithms for Spoken Language Understanding. *In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech).*

[78] Christian Raymond and Giuseppe Riccardi. 2008. Learning with Noisy Supervision for Spoken Language Understanding. *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[79] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. *In Proceedings of the third Text REtrieval Conference (TREC)*.

[80] Gerard Salton and Chris Buckley. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management.* 24 (5): 513–523.

[81] Sunita Sarawagi and William W. Cohen. 2004. Semi-Markov Conditional Random Fields for Information Extraction. *In Advances in Neural Information Processing Systems (NIPS).*

[82] Christina Sauper, Aria Haghighi, and Regina Barzilay. 2010. Incorporating Content Structure into Text Analysis Applications. *In Proceedings of EMNLP*.

[83] Stephanie Seneff. 1992a. TINA: A Natural Language System for Spoken Language Applications. *Computational Linguistics*, 18(1):61--86.

[84] Stephanie Seneff. 1992b. Robust Parsing for Spoken Language Systems. *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[85] Stephanie Seneff. 2007. Web-based Dialogue and Translation Games for Spoken Language Learning. *In Proceedings of the Speech and Language Technology in Education (SLaTE) Workshop.*

[86] Stephanie Seneff and Joseph Polifroni. 2000. Dialogue Management in the Mercury Flight Reservation System. *In Proceedings of Dialogue Workshop, ANLP-NAACL.*

[87] Benjamin Snyder and Regina Barzilay. 2007. Multiple Aspect Ranking using the Good Grief Algorithm. *In Proceedings of the Joint Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies (NAACL-HLT)*.

[88] Amanda J. Stent. 1999. Content Planning and Generation in Continuous-Speech Spoken Dialog. *In Proceedings of KI workshop.*

[89] Ivan Titov and Ryan McDonald. 2008a. Modeling Online Reviews with Multi-Grain Topic Models. *In Proceedings of the 17h International Conference on World Wide Web (WWW)*.

[90] Ivan Titov and Ryan McDonald. 2008b. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[91] G. Tur, A. Stolcke, L. Voss, J. Dowding, B. Favre, R. Fern, M. Frampton, M. Frandsen, C. Frederickson, M. Graciarena, D. Hakkani-tür, D. Kintzing, K. Leveque, S. Mason, J. Niekrasz, S. Peters, M. Purver, K. Riedhammer, E. Shriberg, J. Tien, D. Vergyri, and F. Yang. 2008. The CALO Meeting Speech Recognition and Understanding System. *In Proceedings of Spoken Language Technology Workshop (SLT).*

[92] Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Sentiment Orientation Applied to Unsupervised Classification of Reviews. *In Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).*

[93] Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi, Alexei V. Ivanov, and Pierluigi Roberti. 2009. Leveraging POMDPs trained with User Simulations and Rule-based Dialogue Management in a Spoken Dialogue System. *In Proceedings of SIGDIAL meeting on Discourse and Dialogue*.

[94] Wolfgang Wahlster. 2006. Dialogue Systems Go Multimodal: The SmartKom Experience. *In SmartKom: Foundations of Multimodal Dialogue Systems. Springer*, pages 3-27.

[95] Chao Wang, James Glass, Helen Meng, Joe Polifroni, Stephanie Seneff, and Victor Zue. 1997. YINHE: A Mandarin Chinese Version of the Galaxy System. *In Proceedings of European Conference on Speech Communication and Technology (Eurospeech)*, pages 351-354.

[96] Hsiao-Chuan Wang, Frank Seide, Chiu-Yu Tseng, Lin-Shan Lee. 2000. MAT2000-Design, Collection, and Validation of a Mandarin 2000-Speaker Telephone Speech Database. *In Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 460-463.

[97] Ye-Yi Wang, Alex Acero, Ciprian Chelba, Brendan Frey, and Leon Won. 2002. Combination of Statistical and Rule-Based Approaches for Spoken Language Understanding. *In Proceedings of the International Conference on Spoken Language Processing*, pages 609-612.

[98] Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken Language Understanding — an Introduction to the Statistical Framework. *In IEEE Signal Processing Magazine*.

[99] Ye-Yi Wang, Raphael Hoffmann, Xiao Li, and Jakub Szymanski. 2009. Semi-Supervised Learning of Semantic Classes for Query. Understanding – from the Web and for the Web. *In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM).*

[100] Fuliang Weng, Sebastian Varges, Badri Raghunathan, Florin Ratiu, Heather Pon-barry, Brian Lathrop, Qi Zhang, Harry Bratt, Tobias Scheideck, Kui Xu, Matthew Purver, and Rohit Mishra. 2006. CHAT: A Conversational Helper for Automotive Tasks. *In Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech).*

[101] Janyce M. Wiebe, Rebecca F. Bruce, and Thomas P. O'Hara. 1999. Development and use of a gold standard data set for subjectivity classifications. *In Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 246–253.

[102] Janyce M. Wiebe and Ellen Riloff. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. *In Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing.*

[103] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

[104] Xuehui Wu, 2005. On the Scope of Negation in English, *Sino-US English Teaching*, 2(9):53-56.

[105] Yushi Xu, Jingjing Liu, and Stephanie Seneff. 2008. Mandarin Language Understanding in Dialogue Context. *In Proceedings of International Symposium on Chinese Spoken Language Processing (ISCSLP).*

[106] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. *In Proceedings of the 3rd IEEE international conference on data mining (ICDM).*

[107] Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. *In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP).*

[108] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie Review Mining and Summarization. *In Proceedings of the 15th ACM International Conference on Information and Knowledge Management.*

[109] Victor Zue and James Glass. 2002. Conversational Interfaces: Advances and Challenges. *In Proceedings of the IEEE*, 88(8):1166–1180.

[110] Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen, and Lee Hetherington. 2000. JUPITER: A Telephone-Based Conversational Interface for Weather Information. *In IEEE Transactions on Speech and Audio Processing*, 8(1):85-96.

Appendix A

List of Adjective Sentiment Scores Learned from Our Corpus

Adjective	Sentiment Rating	Frequency
accessible	*4.127752*	*29*
accommodating	*4.177465*	*51*
accurate	*4.667926*	*7*
actual	*2.766121*	*6*
addictive	*4.438897*	*10*
adequate	*2.633585*	*17*
adorable	*4.15737*	*14*
affordable	*4.147568*	*470*
airy	*3.779726*	*10*
amateur	*1.700857*	*9*
amazing	*4.561083*	*991*
ambient	*2.794247*	*8*
ample	*4.01801*	*163*
angry	*1.711221*	*7*
annoying	*1.989833*	*13*
apathetic	*1.987978*	*11*
appropriate	*4.999999*	*7*
arrogant	*1.602638*	*45*
artistic	*4.319414*	*12*
atrocious	*1.543406*	*13*
attentive	*4.11048*	*546*
attractive	*3.760935*	*47*
authentic	*4.318416*	*854*
available	*3.801491*	*68*
average	*2.432339*	*277*
awesome	*4.466628*	*673*
awful	*1.613827*	*295*
awesome	*4.791637*	*37*
bad	*1.679098*	*2549*
baked	*3.785639*	*28*
basic	*2.76238*	*15*
beautiful	*4.032082*	*587*

best	*4.301191*	*1345*
better	*2.700216*	*80*
big	*3.574276*	*394*
bitter	*1.663005*	*6*
black	*3.009035*	*18*
bland	*2.051872*	*388*
bloody	*3.783935*	*16*
blue	*3.797352*	*16*
boring	*2.099764*	*105*
bright	*2.997895*	*29*
brilliant	*4.999999*	*8*
broad	*3.771661*	*12*
brown	*3.386235*	*9*
bubble	*2.863504*	*10*
burnt	*1.919552*	*19*
business	*3.308142*	*8*
bustling	*4.401363*	*12*
busy	*3.108785*	*60*
calm	*4.295297*	*9*
canned	*1.914373*	*8*
careless	*1.557029*	*10*
caring	*3.923812*	*25*
casual	*3.962526*	*277*
central	*3.523575*	*27*
certain	*2.458094*	*9*
charm	*4.728396*	*9*
charming	*4.119469*	*146*
cheap	*4.079543*	*5734*
cheaper	*3.228081*	*9*
cheerful	*4.062103*	*10*
cheesy	*1.982899*	*19*
chic	*3.785006*	*23*
classic	*3.787964*	*43*
classy	*4.202707*	*57*
clean	*4.003149*	*641*
close	*3.184905*	*16*
clueless	*1.705669*	*17*
coconut	*3.267283*	*10*
cold	*1.912673*	*231*
colorful	*4.194696*	*17*
comfortable	*4.158565*	*340*
comfy	*4.098701*	*65*

complex	*3.43848*	*8*
complimentary	*3.627198*	*28*
condescending	*1.960896*	*11*
consistent	*4.144444*	*183*
convenient	*3.699496*	*230*
cooked	*3.352533*	*39*
cool	*3.863299*	*335*
courteous	*4.177835*	*88*
cozy	*4.10562*	*507*
cramped	*2.033908*	*30*
crappy	*1.8371*	*31*
crazy	*2.140211*	*14*
creamy	*3.474998*	*12*
creative	*4.244285*	*314*
creepy	*1.197384*	*9*
crispy	*4.161341*	*22*
crowded	*2.253542*	*431*
crunchy	*3.450192*	*7*
culinary	*2.331497*	*10*
cute	*3.714065*	*237*
daily	*4.098087*	*36*
dead	*2.08881*	*9*
decent	*3.381635*	*330*
decorated	*3.649393*	*17*
deep	*3.893223*	*24*
definite	*3.28355*	*6*
delectable	*4.999999*	*15*
delicious	*4.421734*	*2607*
delightful	*4.679813*	*44*
dependable	*4.185686*	*20*
different	*3.902299*	*102*
difficult	*2.365404*	*27*
dim	*3.512289*	*77*
dingy	*2.238443*	*11*
dining	*3.266267*	*35*
dirty	*1.71884*	*202*
disappointing	*2.026734*	*44*
disgusting	*1.245337*	*59*
dishonest	*1.410218*	*32*
disorganized	*1.855617*	*38*
disrespectful	*1.504108*	*16*
diverse	*4.058596*	*102*

divine	*4.999999*	*19*
double	*3.202194*	*11*
drunken	*3.966434*	*8*
dry	*2.036796*	*65*
dull	*1.932416*	*19*
easy	*3.986937*	*687*
eclectic	*4.251909*	*47*
economical	*3.893874*	*12*
educated	*4.085791*	*8*
efficient	*4.316871*	*77*
elegant	*4.277139*	*95*
empty	*2.152668*	*31*
energetic	*3.877713*	*18*
enjoyable	*4.455953*	*21*
enormous	*3.728657*	*13*
entertaining	*4.180795*	*51*
entire	*1.93787*	*36*
even	*1.802523*	*9*
everyday	*3.893874*	*6*
excellent	*4.442125*	*3480*
exceptional	*4.417982*	*117*
exciting	*4.252507*	*29*
exotic	*3.796683*	*46*
expansive	*2.958602*	*8*
expensive	*2.023778*	*797*
experienced	*3.38711*	*12*
exquisite	*4.643098*	*51*
extensive	*3.980748*	*140*
extra	*2.270987*	*13*
extraordinary	*3.595209*	*17*
fab	*4.114112*	*34*
fabulous	*4.444045*	*218*
fair	*3.513016*	*152*
fake	*1.780031*	*15*
famous	*4.251065*	*6*
fancy	*3.212724*	*11*
fantastic	*4.432559*	*429*
fast	*4.012056*	*1546*
fat	*2.388813*	*10*
fatty	*2.239626*	*9*
favorite	*4.834119*	*16*
festive	*3.756835*	*24*

filthy	*1.745827*	*31*
fine	*3.617871*	*62*
finest	*4.029325*	*6*
first-class	*4.999999*	*7*
fixed	*3.657549*	*10*
flat	*2.432439*	*13*
flavored	*2.541307*	*6*
flavorful	*4.033637*	*156*
flavorless	*1.774826*	*44*
flawless	*4.786667*	*12*
flexible	*4.800906*	*26*
free	*4.018355*	*577*
frequent	*3.308467*	*7*
fresh	*4.225312*	*2391*
freshest	*4.648357*	*52*
fried	*3.528919*	*73*
friendly	*4.209455*	*4721*
front	*2.593411*	*14*
full	*3.838411*	*125*
fun	*4.083922*	*1176*
funky	*3.813523*	*73*
funny	*3.172781*	*13*
fusion	*3.353316*	*21*
gay	*4.626329*	*6*
general	*2.724304*	*44*
generic	*2.024955*	*12*
generous	*4.206902*	*146*
genuine	*4.70119*	*24*
giant	*4.162036*	*11*
good	*3.877736*	*11324*
gorgeous	*4.085794*	*64*
gourmet	*4.294507*	*50*
gracious	*4.046453*	*34*
greasy	*2.094489*	*105*
great	*4.336414*	*20379*
greatest	*4.824352*	*15*
green	*3.535597*	*38*
gross	*1.446858*	*44*
happy	*3.735564*	*175*
hard	*2.22334*	*37*
healthy	*4.189606*	*293*
hearty	*3.947575*	*34*

heated	*3.633681*	*9*
heavenly	*4.424346*	*14*
heavy	*2.473516*	*14*
helpful	*4.073623*	*133*
high	*2.156213*	*171*
highest	*4.626329*	*6*
hip	*3.830173*	*95*
historic	*4.042918*	*18*
homemade	*4.510415*	*113*
home-style	*3.535345*	*10*
homey	*4.288966*	*36*
honest	*4.141561*	*12*
horrendous	*1.767497*	*25*
horrible	*1.540311*	*721*
horrid	*1.633025*	*20*
horrific	*1.277988*	*15*
hospitable	*4.500565*	*17*
hostile	*1.481508*	*17*
hot	*3.555412*	*304*
huge	*4.130165*	*437*
hungry	*2.226396*	*12*
hyped	*2.076852*	*15*
iced	*3.82074*	*14*
ideal	*3.816787*	*12*
ill	*1*	*7*
imaginative	*3.978302*	*20*
impeccable	*4.579213*	*68*
impolite	*1.911552*	*7*
imported	*4.667926*	*7*
impressive	*3.986335*	*21*
inattentive	*2.13259*	*62*
inauthentic	*2.015189*	*10*
incompetent	*1.716528*	*34*
inconsiderate	*1.329692*	*15*
inconsistent	*2.292004*	*70*
incredible	*4.503812*	*247*
indifferent	*1.998516*	*57*
individual	*5*	*6*
indoor	*3.589186*	*10*
inedible	*1.431709*	*30*
inept	*1.692162*	*15*
inexperienced	*2.007632*	*23*

informal	*3.815392*	*11*
informative	*4.728396*	*9*
informed	*4.572818*	*10*
innovative	*4.285827*	*105*
insane	*1.736067*	*8*
inspired	*4.429962*	*13*
intelligent	*3.828317*	*8*
interesting	*3.691425*	*255*
interior	*3.075261*	*77*
intimate	*4.247412*	*257*
inventive	*3.949323*	*82*
juicy	*4.277102*	*25*
kind	*3.805455*	*42*
knowledgeable	*4.313219*	*69*
knowledgeable	*4.376956*	*88*
kosher	*4.185686*	*20*
lackluster	*2.011416*	*21*
lame	*1.656233*	*28*
large	*3.837222*	*763*
larger	*3.38711*	*12*
last	*2.557085*	*6*
late	*3.289414*	*58*
lazy	*1.825471*	*22*
less	*2.530137*	*19*
light	*3.750003*	*8*
lime	*3.54914*	*8*
limited	*2.290739*	*85*
little	*2.385417*	*51*
live	*3.954107*	*271*
lively	*3.808733*	*120*
local	*3.869183*	*47*
long	*2.129111*	*312*
loud	*2.217351*	*249*
lounge	*4.087456*	*44*
lousy	*1.693887*	*163*
lovely	*4.004431*	*157*
low	*3.230539*	*269*
lukewarm	*1.78814*	*9*
magnificent	*4.728396*	*9*
main	*2.698022*	*28*
major	*2.062754*	*6*
marginal	*2.068544*	*14*

marvelous	*4.019397*	*8*
mean	*1.552823*	*36*
mediocre	*2.048913*	*432*
mellow	*4.395152*	*16*
memorable	*3.43757*	*15*
midtown	*3.589975*	*10*
mini	*3.42113*	*11*
miserable	*1.653518*	*15*
misleading	*1.417052*	*15*
mixed	*3.463093*	*19*
moderate	*4.022997*	*84*
modern	*3.746667*	*38*
moist	*3.558821*	*8*
nasty	*1.527871*	*97*
natural	*4.648353*	*13*
nearby	*3.522321*	*6*
neat	*3.811842*	*29*
new	*2.816644*	*124*
nice	*3.688786*	*2129*
noisy	*2.215798*	*174*
nonsmoking	*4.60662*	*28*
noodle	*3.379895*	*15*
north	*3.577215*	*7*
nostalgic	*3.880814*	*13*
numerous	*4.728396*	*9*
obnoxious	*1.368132*	*37*
oily	*1.964601*	*14*
okay	*2.829623*	*18*
old	*2.635264*	*116*
older	*2.651269*	*6*
online	*3.22768*	*13*
open	*3.0079*	*52*
ordinary	*2.329814*	*15*
organic	*3.868563*	*85*
original	*4.11253*	*79*
other	*2.750386*	*74*
outdoor	*3.862821*	*418*
outrageous	*1.55641*	*14*
outstanding	*4.454297*	*275*
overall	*2.821893*	*95*
overcooked	*2.06004*	*45*
overcrowded	*2.045579*	*21*

overpriced	*1.892509*	*859*
overrated	*1.975983*	*67*
own	*2.789471*	*28*
pasta	*3.360142*	*147*
pastrami	*3.966434*	*8*
peaceful	*4.572817*	*15*
perfect	*4.35514*	*228*
personable	*4.406189*	*42*
personal	*4.107487*	*73*
phenomenal	*4.396994*	*22*
plain	*2.319574*	*14*
plastic	*2.295516*	*7*
pleasant	*3.69643*	*168*
pleasing	*4.085791*	*8*
plentiful	*3.859299*	*27*
plenty	*2.812864*	*15*
polite	*3.724934*	*65*
poor	*1.810221*	*1519*
popular	*3.716535*	*14*
positive	*2.557085*	*6*
possible	*1.812553*	*7*
prepared	*3.071839*	*31*
pretentious	*1.838499*	*54*
pretty	*3.5142*	*70*
pricey	*2.378219*	*135*
pricy	*2.340733*	*20*
prime	*3.900933*	*52*
private	*3.9956*	*81*
pro	*3.769145*	*19*
professional	*3.973736*	*122*
prompt	*3.910586*	*337*
public	*4.401363*	*15*
pure	*2.2095*	*7*
pushy	*1.839167*	*31*
quaint	*3.978068*	*71*
quality	*3.576923*	*1091*
questionable	*2.01259*	*19*
quick	*3.939402*	*894*
quiet	*3.982223*	*330*
quirky	*3.966434*	*16*
rare	*4.162036*	*11*
raw	*2.898049*	*37*

real	*3.624088*	*178*
reasonable	*4.12736*	*933*
red	*3.462242*	*23*
refreshing	*4.019397*	*8*
regular	*2.887537*	*7*
relaxed	*4.247422*	*229*
relaxing	*4.27433*	*127*
reliable	*3.796487*	*75*
responsive	*4.539872*	*9*
rich	*4.296233*	*42*
ridiculous	*1.443722*	*17*
right	*3.877032*	*69*
roast	*3.960411*	*7*
romantic	*4.216328*	*978*
roomy	*4.218359*	*11*
round	*3.738106*	*7*
rude	*1.596318*	*1203*
rudest	*1.179386*	*10*
rustic	*4.401363*	*12*
safe	*3.579057*	*13*
salty	*2.175891*	*38*
same	*2.950781*	*16*
satisfying	*3.953372*	*25*
savory	*4.238351*	*15*
scary	*1.981077*	*7*
scenic	*3.966434*	*8*
scrumptious	*4.770236*	*22*
seasonal	*4.114168*	*35*
secluded	*4.999999*	*10*
second	*3.542034*	*8*
secret	*2.968155*	*9*
select	*2.937171*	*8*
separate	*3.413871*	*7*
serious	*2.58328*	*10*
sesame	*3.827619*	*15*
sexy	*4.581794*	*42*
shady	*1.80946*	*15*
short	*3.057144*	*41*
sick	*1.245661*	*26*
simple	*3.784032*	*77*
single	*3.43721*	*7*
skimpy	*2.041314*	*24*

sloppy	*1.84106*	*23*
slow	*2.100594*	*558*
small	*2.368235*	*554*
smaller	*2.28376*	*7*
smart	*4.349114*	*13*
smelly	*1.791254*	*14*
smokey	*2.215856*	*15*
smoking	*2.735722*	*13*
smoky	*2.089056*	*22*
smooth	*4.685435*	*15*
snobby	*1.917567*	*39*
snooty	*2.014166*	*25*
snotty	*2.118898*	*29*
social	*3.183452*	*11*
soft	*3.20032*	*12*
soggy	*1.865374*	*27*
solid	*3.778816*	*63*
soothing	*4.438897*	*10*
sophisticated	*4.560222*	*24*
sour	*2.931489*	*14*
south	*3.520207*	*6*
spacious	*4.023352*	*77*
special	*2.9771*	*65*
specialty	*4.175619*	*26*
spectacular	*3.791244*	*37*
speedy	*3.948087*	*57*
spicy	*4.000273*	*94*
sports	*3.455582*	*16*
stale	*1.88333*	*51*
standard	*2.939954*	*11*
steamed	*3.9232*	*8*
stellar	*3.629839*	*19*
sticky	*2.071219*	*16*
stiff	*3.409267*	*17*
stingy	*2.187164*	*6*
strange	*2.386154*	*12*
strawberry	*4.358447*	*8*
strong	*3.756326*	*68*
stuffed	*3.790673*	*14*
stuffy	*1.979072*	*23*
stunning	*3.989884*	*14*
stupid	*2.069452*	*9*

stylish	*3.831385*	*27*
subpar	*1.910708*	*27*
substandard	*1.735711*	*11*
succulent	*4.551404*	*14*
sucked	*1.877622*	*13*
sunny	*3.518744*	*10*
super	*4.366122*	*79*
superb	*4.624972*	*187*
superior	*4.088881*	*32*
surly	*1.793346*	*22*
sweet	*3.534009*	*57*
tacky	*1.812092*	*20*
talented	*4.551404*	*14*
tasteful	*4.842857*	*17*
tasteless	*1.708597*	*150*
tastiest	*4.601361*	*11*
tasty	*4.001854*	*1391*
tender	*4.4389*	*30*
terrible	*1.558215*	*880*
terrific	*4.420841*	*124*
theatre	*3.797594*	*9*
thick	*3.386975*	*13*
thin	*2.7375*	*11*
thoughtful	*4.667927*	*7*
tight	*2.36093*	*20*
timely	*3.302952*	*21*
tiny	*2.137201*	*58*
tired	*2.027136*	*11*
together	*2.059809*	*6*
top	*4.311064*	*82*
total	*2.326327*	*8*
tough	*2.321285*	*20*
touristy	*2.470427*	*6*
traditional	*3.785759*	*52*
tremendous	*4.999999*	*13*
trendy	*3.424296*	*91*
true	*4.414422*	*46*
typical	*3.096186*	*10*
ugly	*1.528644*	*23*
unacceptable	*1.371157*	*13*
unappetizing	*1.282372*	*12*
unattentive	*2.015005*	*26*

unbeatable	*4.864302*	*20*
unbelievable	*4.584802*	*32*
uncaring	*1.790449*	*12*
unclean	*1.575742*	*18*
uncomfortable	*2.14153*	*38*
uncrowded	*3.930677*	*18*
undercooked	*2.020551*	*22*
uneven	*2.359189*	*11*
unfair	*1.495892*	*6*
unfriendly	*1.854412*	*84*
unhealthy	*1.514509*	*7*
unhelpful	*1.764128*	*15*
unimaginative	*1.901595*	*12*
unimpressive	*2.024955*	*12*
uninspired	*2.091857*	*21*
uninspiring	*2.246072*	*6*
uninteresting	*2.026034*	*14*
unique	*4.187246*	*537*
unlimited	*4.282771*	*21*
unorganized	*2.158245*	*12*
unpleasant	*1.845366*	*16*
unpretentious	*4.326077*	*60*
unprofessional	*1.590055*	*64*
unsanitary	*1.446035*	*15*
unusual	*3.869685*	*56*
unwelcoming	*1.61217*	*11*
upbeat	*4.601361*	*22*
upscale	*3.751157*	*37*
upstairs	*4.02335*	*7*
validated	*4.145963*	*44*
varied	*3.96627*	*93*
vegan	*3.85993*	*49*
vegetarian	*3.422101*	*89*
waiting	*2.203978*	*13*
warm	*3.838928*	*329*
weak	*1.982441*	*43*
weird	*2.226488*	*28*
welcome	*3.661326*	*21*
welcoming	*4.505882*	*74*
well	*3.747084*	*43*
white	*3.672826*	*16*
whole	*2.653758*	*51*

wide	*3.97213*	*123*
wild	*2.467383*	*9*
wireless	*3.564557*	*14*
wonderful	*4.484477*	*798*
worse	*1.546163*	*22*
worst	*1.441473*	*180*
wrong	*1.666056*	*32*
young	*3.051081*	*20*
yucky	*1.440391*	*14*
yum	*4.074152*	*73*
yummy	*4.226891*	*572*

Appendix B

List of Context-Relevant Topics Extracted from Our Corpus

Topic	Grammar Rules
experience	*experience :*active* "1"*
music	*music :*active* "1"*
air	*air :*active* "1"*
booths	*booths :*active* "1"*
chairs	*chairs :*active* "1"*
dates	*dates :*active* "1"*
decorations	*decorations :*active* "1"*
dinners	*dinners :*active* "1"*
door	*door :*active* "1"*
energy	*energy :*active* "1"*
environment	*environment :*active* "1"*
family_atmosphere	*family_atmosphere :*active* "1"*
feeling	*feeling :*active* "1"*
jazz_music	*jazz_music :*active* "1"*
lounge	*lounge :*active* "1"*
mood	*mood :*active* "1"*
neighborhood_place	*neighborhood_place :*active* "1"*
neighborhood_spot	*neighborhood_spot :*active* "1"*
patio	*patio :*active* "1"*
rooms	*rooms :*active* "1"*
scene	*scene :*active* "1"*
setting	*setting :*active* "1"*
venue	*venue :*active* "1"*
views	*views :*active* "1"*
vibe	*vibe :*active* "1"*
area	*area :*active* "1"*
bakery	*bakery :*active* "1"*
band	*band :*active* "1"*
bar	*bar :*active* "1"*
bar_area	*bar_area :*active* "1"*
bar_scene	*bar_scene :*active* "1"*

bathrooms	*bathrooms :*active* "1"*
bistro	*bistro :*active* "1"*
booth	*booth :*active* "1"*
breakfast_place	*breakfast_place :*active* "1"*
breakfast_spot	*breakfast_spot :*active* "1"*
building	*building :*active* "1"*
coffee_shop	*coffee_shop :*active* "1"*
crowd	*crowd :*active* "1"*
crowds	*crowds :*active* "1"*
date_spot	*date_spot :*active* "1"*
dining_room	*dining_room :*active* "1"*
district	*district :*active* "1"*
garden	*garden :*active* "1"*
location	*location :*active* "1"*
look	*look :*active* "1"*
lunch_spot	*lunch_spot :*active* "1"*
neighborhood	*neighborhood:*active* "1"*
neighborhood_bar	*neighborhood_bar :*active* "1"*
neighborhood_restaurant	*neighborhood_restaurant :*active* "1"*
parking_lot	*parking_lot :*active* "1"*
pizza_place	*pizza_place :*active* "1"*
pub	*pub :*active* "1"*
restaurant	*restaurant :*active* "1"*
restaurants	*restaurants :*active* "1"*
salad_bar	*salad_bar :*active* "1"*
space	*space :*active* "1"*
sports_bar	*sports_bar :*active* "1"*
spot	*spot :*active* "1"*
spots	*spots :*active* "1"*
steak_house	*steak_house :*active* "1"*
surrounding	*surrounding :*active* "1"*
sushi_spot	*sushi_spot :*active* "1"*
sushi_place	*sushi_place :*active* "1"*
table	*table :*active* "1"*
tables	*tables :*active* "1"*
vietnamese_restaurant	*vietamese_restaurant :*active* "1"*
wine_bar	*wine_bar :*active* "1"*
wine_pairings	*wine_parings :*active* "1"*