

Amy Neustein
Editor

Advances in Speech Recognition

Mobile Environments,
Call Centers and Clinics

Foreword by
Judith Markowitz and Bill Scholz



Advances in Speech Recognition

Amy Neustein
Editor

Advances in Speech Recognition

Mobile Environments, Call Centers
and Clinics

Foreword by
Judith Markowitz and Bill Scholz



Springer

Editor

Amy Neustein
Linguistic Technology Systems
Fort Lee, New Jersey
USA
amy.neustein@verizon.net

ISBN 978-1-4419-5950-8 e-ISBN 978-1-4419-5951-5

DOI 10.1007/978-1-4419-5951-5

Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010935485

© Springer Science+Business Media, LLC 2010

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Two Top Industry Leaders Speak Out

Judith Markowitz

When Amy asked me to co-author the foreword to her new book on advances in speech recognition, I was honored. Amy's work has always been infused with creative intensity, so I knew the book would be as interesting for established speech professionals as for readers new to the speech-processing industry.

The fact that I would be writing the foreword with Bill Scholz made the job even more enjoyable. Bill and I have known each other since he was at UNISYS directing projects that had a profound impact on speech-recognition tools and applications.

Bill Scholz

The opportunity to prepare this foreword with Judith provides me with a rare opportunity to collaborate with a seasoned speech professional to identify numerous significant contributions to the field offered by the contributors whom Amy has recruited.

Judith and I have had our eyes opened by the ideas and analyses offered by this collection of authors. Speech recognition no longer needs be relegated to the category of an experimental future technology; it is here today with sufficient capability to address the most challenging of tasks. And the point-click-type approach to GUI control is no longer sufficient, especially in the context of limitations of modern-day hand held devices. Instead, VUI and GUI are being integrated into unified multimodal solutions that are maturing into the fundamental paradigm for computer-human interaction in the future.

Judith Markowitz

Amy divided her book into three parts but the subject of the first part, mobility, is a theme that flows through the entire book – which is evidence of the extent to which mobility permeates our lives. For example, Matt Yuschik's opening chapter

in the Call Centers section, which makes up the second part of the book, considers the role of multimodality for supporting mobile devices.

Accurate and usable mobile speech has been a goal that many of us have had for a long time. When I worked for truck-manufacturer Navistar International in the 1980s, we wanted to enable drivers to perform maintenance checks on-the-fly by issuing verbal commands to a device embedded in the truck. At that time, a deployment like that was a dream. Chapters in all three sections of this book reveal the extent to which that dream has been realized – and not just for mobile phones. For example, James Rodger and James George’s chapter in the Clinics section examines end-user acceptance of a handheld, voice-activated device for preventive healthcare.

Bill Scholz

The growing availability of sophisticated mobile devices has stimulated a significant paradigm shift resulting from a combination of sophisticated speech capability with limited graphic input and display capability. The need for a paradigm shift is exacerbated by the increased frequency with which applications formerly constrained to desktop computers migrate onto mobile devices, only to frustrate users accustomed to click-and-type input and extensive screen real estate output. Bill Meisel’s introductory chapter brings this issue into clear focus, and Mike Phillips’ team offers candidate solutions in which auditory and visual cues are augmented by tactile and haptic feedback to yield multimodal interfaces which overcome many mobile device limitations.

In response to the demand for more accurate speech input on mobile devices, Mike Cohen’s team from Google has enhanced every step of the recognition process, from text normalization and acoustic model development through language model training using billions of words. Sophisticated endpointing permits removal of press-to-talk keys, and in collaboration with enhanced multimodal dialog design, provides a comfortable conversational interface that is a natural extension to traditional Web access.

Sid-Ahmed Selouani summarizes efforts of the European community to enhance both the input phase of speech recognition through techniques such as line spectral frequency analysis, and the use of an AI markup language to facilitate interpretation of recognizer output.

The chapters in the Call Centers section describe an array of technologies. Matt Yusich shows us data justifying the importance of multimodality in contact centers to facilitate caller–agent communication. The combination of objective and subjective measures identified by Roberto Pieraccini’s team provides metrics for contact center evaluation that dramatically reflects the communication performance enhancements that result from the increased emphasis on multimodal dialog.

Judith Markowitz

Emphasizing the importance of user expectations, Stephen Springer delves deeply into the subjective aspects of user interface design for call centers. This chapter is a tremendous resource for designers whether they are working with speech for the first time or seasoned developers.

Good design is important but problem dialogs can occur even when callers interact with well-designed speech systems or human agents. Unlike many emotion-detection systems, the tool that Alexander Schmitt and his co-authors have constructed for detecting anger and frustration is not limited to acoustic indicators; it also analyzes words, phrases, the dialog as a whole, and prior emotional states.

While Alexander Schmitt and his co-authors focus on resolving problem dialogs for individual callers, Marsal Gavalda and Jeff Schlueter address problems that occur at the macro level. They describe a phonetics-based, speech-analytics system capable of indexing more than 30,000 h of a contact center's audio and audio-visual data in a single day and then mining the index for business intelligence.

I was pleased to see a section on speech in clinical settings. John Shagoury crafted a fine examination of medical dictation that shows why speech recognition has become an established and widely accepted method for generating medical reports.

Most treatments of speech recognition in clinics rarely go much beyond its use for report generation. Consequently, I was happy to see chapters on a portable medical device and on the use of speech and language for diagnosis and treatment. Julia Hirschberg and her co-authors' literature review demonstrates that not only are there acoustic and linguistic indicators of diseases as disparate as depression, diabetes, and cancer but also that some of those indicators can be used to measure the effectiveness of treatment regimens. Similarly, Hemant Patil's classification of infant cries gives that population of patients a "voice" to communicate about what is wrong. If I had had such tools when I worked as a speech pathologist in the 1970s, I would have been able to do far more for the betterment of my patients.

Amy Neustein has compiled an excellent overview of speech for mobility, call centers, and clinics. Bravo!

Judith Markowitz, Ph.D., is president of J. Markowitz Consultants, and is recognized internationally as one of the top analysts in speech processing. For over 25 years, she has provided strategic and technical consulting to large and small organizations, and has been actively involved in the development of standards in biometrics and speech processing. In 2003, she was voted one of the top ten leaders in the speech-processing industry and, in 2006, she was elevated to IEEE Senior Member status. Among Dr. Markowitz's many accomplishments, she served with distinction as technology editor of *Speech Technology Magazine* and chaired the VoiceXML Forum Speaker Biometrics Committee.

K.W. “Bill” Scholz, Ph.D., is the president of AVIOS, the speech industry’s oldest professional organization. He founded NewSpeech, LLC in 2006, following his long tenure at Unisys, where he served as Director of Engineering for Natural Language solutions. His long and distinguished career as a consultant for domestic and international organizations in architectural design, speech technology, knowledge-based systems, and integration strategies is focused on speech application development methodology, service creation environments, and technology assessment.

Preface

Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics provides a forum for today's speech technology industry leaders – drawn from private enterprises and academic institutions all over the world – to discuss the challenges, advances, and aspirations of voice technology.

The collection of essays contained in this volume represents the research findings of over 30 speech experts, including speech engineers, system designers, linguists, and IT (information technology) and MIS (management information systems) specialists. The book's 14 chapters are divided into three sections – mobile environments, call centers, and clinics. But given the practical ubiquity of mobile devices, this three-part division sometimes seems almost irrelevant. For example, one of the chapters in the “call centers” section provides a vivid discussion of how to provide today's call centers with multimodal capabilities – to support text, graphic, voice, and touch – in self-service transactions, so that customers who contact the call center using their mobile phones (rather than a fixed line) can expect a sophisticated interface that lets them resolve their service issues in a way that uses the full capabilities of their handsets, and similarly call center agents using mobile devices that support multimodality can experience more efficient navigation and retrieval of information to complete a transaction for a caller. In the “clinics” section, for that matter, one of the chapters focuses on validating user satisfaction with a voice-activated medical tracking application, run on a compact mobile device for a “hands-free” method of data entry in a clinical setting.

In spite of this unavoidable overlap of sections, the authors' earnest discussions of the manifold aspects of speech technology not only complement one another but also divide into several areas of specific interest. Each author brings to this round-table his or her unique insights and new ideas, the fruits of much time spent formulating, developing and testing out their theories about what kinds of voice applications work best in mobile settings, call centers and clinics.

The book begins with an introduction to the role of speech technology in mobile applications written by Bill Meisel, President of TMA Associates. Meisel is also editor of *Speech Strategy News* and co-chair (with AVIOS) of the annual *Mobile Voice* conference in northern California. He opens his discussion by quoting the predictions published by the financial investment giant Morgan Stanley in its *Mobile Internet Report*, issued near the end of 2009. Meisel shows that in Morgan

Stanley's 694-page report, Mobile Internet Computing was said to be "the technology driver of the next decade," following the Desktop Internet Computing of the 1990s, the Personal Computing of the 1980s, the Mini-Computing of the 1970s and, finally, the Mainframe Computing of the 1960s. In his chapter, fittingly titled "Life on the Go – The Role of Speech Technology in Mobile Applications," Meisel asserts that since "the mobile phone is becoming an indispensable personal communication assistant and multi-functional device... [such a] range of applications creates user interaction issues that can't be fully solved by extending the Graphical User Interface and keyboard to these small devices." "Speech recognition, text-to-speech synthesis, and other speech technologies," Meisel continues, "are part of the solution, particularly since, unlike PCs, every mobile phone has a microphone and speech output."

Advances in Speech Recognition – which is being published at the very beginning of this auspicious decade for mobile computing – examines the practical constraints of using voice in tandem with text. Following Meisel's comprehensive overview of the role of speech technology in mobile applications, Scott Taylor, Vice President of Mobile Marketing and Solutions at Nuance Communications, Inc., offers a chapter titled "Striking a Healthy Balance – Speech Technology in the Mobile Ecosystem." Here, Taylor cautions the reader about the need to "balance a variety of multimodal capabilities so as to optimally fit the user's needs at any given time." While there is "no doubt that speech technologies will continue to evolve and provide a richer user experience," argues Taylor, it is critical for experts to remember that "the key to success of these technologies will be thoughtful integration of these core technologies into mobile device platforms and operating systems, to enable creative and consistent use of these technologies within mobile applications." This is why speech developers, including Taylor himself, view speech capabilities on mobile devices not as a single entity but rather as part of an *entire* mobile ecosystem that must strive to maintain homeostasis so that consumers (as well as carriers and manufacturers) will get the best service from a given mobile application.

To achieve that goal, Mike Phillips, Chief Technology Officer at Boston-based Vlingo, together with members of the company has been at pains to design more effective and satisfying multimodal interfaces for mobile devices. In the chapter following Taylor's, titled "Why Tap When You Can Talk – Designing Multimodal Interfaces for Mobile Devices that Are Effective, Adaptive and Satisfying to the User," Phillips and his co-authors present findings from over 600 usability tests in addition to results from large-scale commercial deployments to augment their discussion of the opportunities and challenges presented in the mobile environment. Phillips and his co-writers stress how important it is to strive for user-satisfaction: "It is becoming clear that as mobile devices become more capable, the user interface is the last remaining barrier to the scope of applications and services that can be made available to the users of these devices. It is equally clear that speech has an important role to play in removing these user interface barriers."

Johan Schalkwyk, Senior Staff Engineer at Google, along with some of his colleagues provide the book's next chapter, aptly titled "Your Word is my

Command –Google Search by Voice: A Case Study.” In this chapter, Schalkwyk and his co-authors illuminate the technology employed by Google “to make search by voice a reality” – and follow this with a fascinating exploration of the user interface side of the problem, which includes detailed descriptions and analyses of the specifically tailored user studies that have been based on Google’s deployed applications.

In painstaking detail, Schalkwyk and his colleagues demystify the complicated technology behind 800-GOOG-411 (an automated system that uses speech recognition and web search to help people find and call businesses), GMM (Google Maps for Mobile) which – unlike GOOG-411 – applies a multimodal speech application (making use of graphics), and finally the Google Mobile application for the iPhone, which includes a search by voice feature. The coda to the chapter is its discussion of user studies based on analyses of live data, and how such studies reveal important facts about user behavior, facts that impact Google’s “decisions about the technology and user interfaces.” Here are the essential questions addressed in those user studies: “What are people actually looking for when they are mobile? What factors influence them to choose to search by voice or type? What factors contribute to user satisfaction? How do we maintain and grow our user base? How can speech make information access easier?”

The mobile environments section concludes with the presentation of a well-planned study on speech recognition in noisy mobile environments. Sid-Ahmed Selouani, Professor of Information Management at the Université de Moncton, Shippagan Campus, New Brunswick, Canada, in “Well Adjusted – Using Robust and Flexible Speech Recognition Capabilities in Clean to Noisy Mobile Environments,” presents study findings on a new speech-enabled framework that aims at providing a rich interactive experience for smartphone users – particularly in mobile environments that can benefit from hands-free and/or eyes-free operations.

Selouani introduces this framework by arguing that it is based on a conceptualization that divides the mapping between the speech acoustical microstructure and the spoken implicit macrostructure into two distinct levels, namely the signal level and linguistic level. At the signal level, a front-end processing that aims at improving the performance of Distributed Speech Recognition (DSR) in noisy mobile environments is performed.

The linguistic level, on the contrary, “involves a dialogue scheme to overcome the limitations of current human-computer interactive applications that are mostly using constrained grammars.” “For this purpose,” says Selouani, “conversational intelligent agents capable of learning from their past dialogue experiences are used.”

In conducting this research on speech recognition in clean to noisy mobile environments, Selouani utilized the Carnegie-Mellon Pocket Sphinx engine for speech recognition and the Artificial Intelligence Markup Language (AIML) for pattern matching. The evaluation results showed that including both the Genetic Algorithms (GA)-based front-end processing and the AIML-based conversational agents led to significant improvements in the effectiveness and performance of an interactive spoken dialog system in a mobile setting.

Matthew Yuschik, Senior User Experience Specialist at Cincinnati-based Convergys Corporation provides the perfect segue to the next section of *Advances in Speech*

Recognition. In “It’s the Best of all Possible Worlds – Leveraging Multimodality To Improve Call Center Productivity,” Yuschik makes a convincing argument for equipping today’s call centers with multimodal capabilities in self-service transactions – to support text, graphic, voice, and touch – so that customers who contact the call center using their mobile phones (rather than a fixed line) can expect an interface that “provides multiple ways for the caller to search for resolution of their [service] issue.” Given market research predictions that there will be over 4 billion wireless subscribers in 2010, Yuschik draws the sound conclusion that more and more callers will be using their mobile devices when availing themselves of customer support services at customer care and contact centers. After all, most customers who need to resolve product and service issues, or to order new products and services, squeeze in their calls “on the go” instead of taking up crucial time while working at their desks.

In “It’s the Best of all Possible Worlds,” Yuschik explains how leveraging multimodality to improve call center productivity is achieved by striking a healthy balance between satisfying the caller’s goal and maximizing the agent’s productivity in the call center. He points out that “a multimodal interface can voice-enable all features of a GUI.” Yet, he cautions “this is a technologically robust solution, but does not necessarily take into account the caller’s goal.” Conceding that “voice activating all parts of the underlying GUI of the application enables the agent to solve every problem by following the step-by-step sequence imposed by the GUI screens,” Yuschik states that “a more efficient approach...is to follow the way agents and callers carry on their dialog to reach the desired goal.” He shows that “this scenario-based (use-case) flow – with voice-activated tasks and subtasks – with tasks and subtasks voice activated – provides a streamlined approach in which an agent follows the caller-initiated dialog, using the MMUI [multimodal user interface] to enter data and control the existing GUI in any possible sequence of steps. This goal-focused view,” as explained by Yuschik, “enables callers to complete their transactions as fast as possible.”

Yuschik’s chapter details a set of Convergys trials that “follow a specific sequence where multimodal building-blocks are identified, investigated, and then combined into support tasks that handle call center transactions.” Crucial to those trials were the Convergys call center agents who “tested the Multimodal User Interface for ease of use, and efficiency in completing caller transactions.” The results of the Convergys trials showed that “multimodal transactions are faster to complete than only using a Graphical User Interface.” Yuschik concludes that “the overarching goal of a multimodal approach should be to create a framework that supports many solutions. Then,” he writes, “tasks within any specific transaction are leveraged across multiple applications.”

Every new technology deserves an accurate method of evaluating its performance and effectiveness; otherwise, the technology will not fully serve its intended purpose. David Suendermann, Principal Speech Scientist at the New York-based SpeechCycle, Inc., and his colleagues Roberto Pieraccini and Jackson Liscombe, are joined by Keelan Evanini of Educational Testing Services in Princeton, New Jersey, for the presentation of an enlightening discussion of a new framework to measure accurately the performance of automated customer care contact centers.

In “‘How am I Doing?’ – A New Framework To Effectively Measure the Performance of Automated Customer Care Contact Centers,” the authors carefully dissect conventional methods of measuring how satisfied customers are with automated customer care and contact centers, pointing out why such methods can produce woefully misleading results. They point to a problem that is ever-present when evaluating callers’ satisfaction with any of these self-service contact centers. Namely: quantifying how effectively interactive voice response (IVR) systems satisfy callers’ goals and expectations “has historically proven to be a most difficult task.” Suendermann and his co-authors convincingly show that

[s]uch difficulties in assessing automated customer care contact centers can be traced to two assumptions [albeit misguided] made by most stakeholders in the call center industry:

1. Performance can be effectively measured by deriving statistics from call logs; and
2. The overall performance of an IVR can be expressed by a single numeric value.

The authors introduce an IVR assessment framework that confronts these misguided assumptions head on, demonstrating how they can be overcome. The authors show how their “new framework for measuring the performance of IVR-driven call centers incorporates objective and subjective measures.” Using the concepts of *hidden* and *observable* measures, the authors demonstrate how to produce metrics that are reliable and meaningful so that they can better provide accurate system design insights into multiple aspects of IVR performance in call centers.

Just as it is possible to jettison poor methods of evaluating caller satisfaction with IVR performance in favor of more accurate ones, it is equally possible to meet (or even exceed) user expectations with the design of a speech-only interface that builds on what users have come to expect from self-service delivery in general, whether at the neighborhood pharmacy or at the international airport. Stephen Springer, Senior Director of User Interface Design at Nuance Communications, Inc., shows how to do this in his chapter (aptly) titled “Great Expectations – Making Use of Callers’ Experiences from Everyday Life To Design a Satisfying Speech-Only Interface for the Call Center.” According to Springer, “the thoughtful use of user modeling achieved by employing ideas and concepts related to transparency, choice, and expert advice, all of which most, if not all, callers are already familiar with from their own everyday experiences” better meets the users’ expectations than systems whose workings are foreign to what such users encounter in day-to-day life.

Springer carefully examines a wide variety of expectations that callers bring to self-service phone calls, ranging from broad expectations about self-service in general to the more specific expectations of human-to-human conversation about consumer issues. As a specialist in user interface design, Springer recommends to the system designer several indispensable steps to produce more successful interaction between callers and speech interfaces. The irony is that the secrets for meeting greater expectations for caller satisfaction with speech-only interfaces in the call center are not really secrets: they can be found uncannily close to home, by

extrapolating from callers' everyday self-service experiences and from their quotidian dialog with human agents at customer care contact centers.

Next, two German academics and SpeechCycle's CTO, Roberto Pieraccini, tackle the inscrutable and often elusive emotions of callers to ascertain when task completion and user satisfaction with the automated call center may be at risk. Alexander Schmitt of Ulm University, and his two co-authors, in their chapter titled “For Heaven’s Sake, Gimme a Live Person!” – Designing Emotion-Detection Customer Care Voice Applications in Automated Call Centers,” show how their voice application can robustly detect angry user turns by considering acoustic, linguistic, and interaction parameter-based information – all of which can be collected and exploited for anger detection. They introduce, in addition, a valuable subcomponent that is able to estimate the emotional state of the caller based on the caller’s *previous* emotional state, supporting the theory that anger displayed in calls to automated call centers, rather than being an isolated occurrence, is more likely an incremental build-up of emotion. Using a corpus of 1,911 calls from an Interactive Voice Response system, the authors demonstrate the various aspects of speech displayed by angry callers.

The call center section of *Advances in Speech Recognition* is rounded off by a fascinating chapter on advanced speech analytic solutions aimed at learning why customers call help-line desks and how effectively they are served by the human agent. Yes, that is correct: a *human* agent, a specimen of call center technology that still exists notwithstanding the push for heavily automated self-service centers. In “The Truth Is Out There – Using Advanced Speech Analytics To Learn Why Customers Call Help-Line Desks and How Effectively They’re Being Served by the Call Center Agent,” Marsal Gavalda, Vice President of Incubation and Principal Language Scientist at Nexidia, and Jeff Schlueter (the company’s Vice President of Marketing & Business Development) describe their novel work in phonetic-based indexing and search, which is designed for extremely fast searching through vast amounts of media.

The authors of “The Truth is Out There” explain the nuts and bolts of their method, showing how they “search for words, phrases, jargon, slang and other terminology that are not readily found in a speech-to-text dictionary.” They demonstrate how “the most advanced phonetic-based speech analytics solutions,” such as theirs, “are those that are robust to noisy channel conditions and dialectal variations; those that can extract information beyond words and phrases; and those that do not require the creation or maintenance of lexicons or language models.” The authors assert that “such well performing speech analytic programs offer unprecedented levels of accuracy, scale, ease of deployment, and an overall effectiveness in the mining of live and recorded calls.” Given that speech analytics has become indispensable to understanding how to achieve a high rate of customer satisfaction and cost containment, Gavalda and his co-author demonstrate in their chapter how their data mining technology is used to produce sophisticated analyses and reports (including visualizations of call category trends and correlations or statistical metrics), while preserving “the ability at any time to drill down to individual calls and listen to the specific evidence that supports the particular categorization or data

point in question, all of which allows for a deep and fact-based understanding of contact center dynamics.”

John Shagoury, Executive Vice President of the Healthcare & Imaging Division of Nuance Communications, Inc., opens *Advances in Speech Recognition*’s last section with a cogent discussion of “the benefits of incorporating speech recognition as part of the everyday clinical documentation workflow.” In his chapter – fittingly titled “Dr. Multi-Task – Using Speech To Build up Electronic Medical Records While Caring for Patients” – Shagoury shows how speech technology yields a significant improvement in the quality of patient care by increasing the speed of the medical documentation process, so that patients’ health records are quickly made available to healthcare providers. This means they can deliver timely and efficient medical care. Using some fascinating, and on point, real-world examples, Shagoury richly demonstrates how the use of speech recognition technology directly affects improved productivity in hospitals, significant cost reductions, and overall quality improvements in the physician’s ability to deliver optimal healthcare. But Shagoury does not stop there. He goes on to demonstrate that “beyond the core application of speech technologies to hospitals and primary care practitioners, speech recognition is a core tool within the diagnostics field of healthcare, with broad adoption levels within the radiology department.”

Next, James Rodger, Professor of Management Information Systems and Decision Sciences at Indiana University of Pennsylvania, Eberly College of Business and Information Technology – with his co-author, James A. George, senior consultant at Sam, Inc. – provides the reader with a rare inside look at the authors’ “decade long odyssey” in testing and validating end-user acceptance of speech in the clinical setting aboard US Navy ships. In their chapter, titled “Hands Free – Adapting the Task-Technology-Fit Model and Smart Data To Validate End-User Acceptance of the Voice Activated Medical Tracking Application (VAMTA) in the United States Military,” the authors show how their extensive work on validating user acceptance of VAMTA – which is run on a compact mobile device that enables a “hands-free” method of data entry in the clinical setting – was broken down into two phases: 1) a pilot to establish validity of an instrument for obtaining user evaluations of VAMTA and 2) an in-depth study to measure the adaptation of users to a voice-activated medical tracking system in preventive health care. For the latter phase, they adapted a task-technology-fit (TTF) model (from a smart data strategy) to VAMTA, demonstrating that “the perceptions of end-users can be measured and, furthermore, that an evaluation of the system from a conceptual viewpoint can be sufficiently documented.” In this chapter, they report on both the pilot and the in-depth study.

Rodger and his co-author applied the Statistical Package for the Social Sciences (SPSS) data analysis tool to analyze the survey results from the in-depth study to determine whether TTF, along with individual characteristics, will have an impact on user evaluations of VAMTA. In conducting this in-depth study, the authors modified the original TTF model to allow adequate domain coverage of patient care applications. What is most interesting about their study – and perhaps a testament to the vision of those at the forefront of speech applications in the clinical setting – is that,

according to Rodger and his co-author, their work “provides the underpinnings for a subsequent, higher level study of nationwide medical personnel.” In fact, they intend “follow-on studies [to] be conducted to investigate performance and user perceptions of VAMTA under *actual* medical field conditions.”

Julia Hirschberg and Noémie Elhadad, distinguished faculty members at Columbia University, in concert with Anna Hjalmarsson, a bright and talented Swedish graduate student studying at KTH (Royal Institute of Technology), make a strong argument that if “language cues” – primarily acoustic signal and lexical and semantic features – “can be identified and quantified automatically, this information can be used to support diagnosis and treatment of medical conditions in clinical settings [as well as] to further fundamental research in understanding cognition.” In “You’re As Sick As You Sound – Using Computational Approaches for Modeling Speaker State To Gauge Illness and Recovery,” Hirschberg and her co-authors perform an exhaustive medical literature review of studies “that explore the possibility of finding speech-based correlates of various medical conditions using automatic, computational methods.” Among the studies they review are computational approaches that explore communicative patterns of patients who suffer from medical conditions such as depression, autism spectrum disorders, schizophrenia, and cancer.

The authors see a ripe opportunity here for future medical applications. They point out that the emerging research into *speaker state* for medical diagnostic and treatment purposes – an outgrowth of “related work on computational modeling of emotional state” for studying callers’ interactions with call center agents and Interactive Voice Response (IVR) applications “for which there is interest in distinguishing angry and frustrated callers from the rest” – equips the physician with a whole new set of diagnostic and treatment tools. “Such tools can have economic and public health benefits, in that a wider population – particularly individuals who live far from major medical centers – can be efficiently screened for a broader spectrum of neurological disorders,” they write. “Fundamental research on mental disorders, like post-partum depression and post traumatic stress disorder, and coping mechanisms for patients with chronic conditions, like cancer and degenerative arthritis, can likewise benefit from computational models of speaker state.”

Hemant Patil, Assistant Professor at the Dhirubhai Ambani Institute of Information and Communication Technology, DA-IICT, in Gandhinagar, India, echoes the beliefs of Shagoury, Rodger and George, and of Hirschberg, Hjalmarsson and Elhadad, all of whom maintain that advances in speech technology have untold economic, social, and public health benefits. In “‘Cry Baby’ – Using Spectrographic Analysis To Assess Neonatal Health Status from an Infant’s Cry,” Patil demonstrates that the rich body of research on spectrographic analysis, predominantly used for performance of speaker recognition, may also be used to assess the neonate’s health status, by comparing a normal to an abnormal cry.

Spectrographic analysis is seen by Patil and his colleagues – who are just as passionately involved in this highly specialized area of infant cry research – as useful in improving and complementing “the clinical diagnostic skills of pediatricians and neonatologists, by helping them to detect early warning signs of pathology,

developmental lags, and so forth.” Patil points out to the reader that such technology “is especially helpful in today’s healthcare environment, in which newborns do not have the luxury of being solely attended by one physician, and are, instead, monitored remotely by a centralized computer control system.”

In explaining cry analysis – a multidisciplinary area of research integrating pediatrics, neurology, physiology, engineering, developmental linguistics, and psychology – Patil demonstrates in “Cry Baby” his application of spectrographic analysis to the vocal sounds of an infant, comparing normal with abnormal infant crying. In his study, ten distinct cry modes, *viz.*, hyperphonation, dysphonation, inhalation, double harmonic break, trailing, vibration, weak vibration, flat, rising, and falling, have been identified for normal infant crying, and their respective spectrographic patterns were observed. This analysis was then extended to the abnormal infant cry. Patil observed that

the *double harmonic break* is more dominant for abnormal infant cry in cases of myalgia (muscular pain). The *inhalation* pattern is distinct for infants suffering from asthma or other respiratory ailments such as a cough or cold. For example, for the infant whose larynx is not well developed, the *pitch harmonics* are nearly absent. As such, there are no voicing or glottal vibrations in the cry signal. And for infants with Hypoxic Ischemic Encephalopathy (HIE), there is an initial tendency of pitch harmonics to rise and then to be followed by a blurring of such harmonics.

As part of this study, Patil also performed infant cry analysis by observing the nature of the optimal warping path in the Dynamic Time Warping (DTW) algorithm, which is found to be “near diagonal” in healthy infants, in contrast to that in unhealthy infants whose warping paths reveal significant deviations from the diagonal across most, though not all, cry modes.

Looking further into broader sociologic implications of cry analysis, Patil shows how this novel field of research can redress the social and economic inequities of healthcare delivery. “Motivated by a need to equalize the level of neonatal healthcare (not every neonate has the luxury of being monitored at a teaching hospital equipped with a high level neonatal intensive care unit), I propose for the next phase of research a quantifiable measurement of the added clinical advantage to the clinician (and ancillary healthcare workers) of a baseline comparison of normal versus abnormal cry.”

Now it is up to the reader, after assimilating the substance of this book, to envision how speech applications in mobile environments, call centers, and clinics will improve the lives of consumers, corporations, carriers, manufacturers, and healthcare providers – to say nothing of the overall improvements that such technology provides for the byzantine social architecture known as modern-day living.

Fort Lee, NJ

Amy Neustein, Ph.D

Acknowledgments

This book would not have been possible without the support and encouragement of Springer's Editorial Director, Alex Greene, and of his editorial assistant, Ciara J. Vincent, and of the production editor, Joseph Quatela, and the project manager, Rajesh Harini, who in the final stages of production attended with much alacrity to each and every detail. Every writer/editor needs an editor and I could not have asked for a more clear-thinking person than Alex Greene. Alex's amazing vision helped to shepherd this project from its inception to fruition.

I remain grateful to Drs. Judith Markowitz and K.W. "Bill" Scholz, who contributed an illuminating foreword to this book, and to Dr. James Larson, whose fascinating look into the future provides a fitting coda to this book. Of equal importance is Dr. Matthew Yusichik, Senior User Experience Specialist at Convergys Corporation, who generously offered to review all three sections of this work, a task that consumed a large portion of his weekends and evenings. I will never be able to sufficiently thank Matt for his astute and conscientious review.

Dr. William Meisel, President of TMA Associates in Tarzana, CA and Editor of *Speech Strategy News*, deserves a special acknowledgment. If there is one person who has his finger on the pulse of the speech industry, it is Bill Meisel. Bill's clarity of thought helped me to see the overarching theme of mobile applications.

Finally, I'd like to acknowledge several of the "foot soldiers" – the principal authors who shouldered the burden of the project. Johan Schalkwyk, Google's Senior Staff Engineer deserves particular thanks for meeting his chapter submission deadline even though he had to work evenings and weekends to do it. Dr. David Suendermann, Principal Speech Scientist at SpeechCycle, Inc. sat dutifully at his desk during a major snowstorm in New York, answering a series of e-mails containing editing queries. Alexander Schmitt, Scientific Researcher at the Institute of Information Technology at Ulm University, worked tirelessly – and often late into the night – to answer my editing queries as quickly as possible notwithstanding the six-h time difference between New York and Germany. And in India, Dr. Hemant Patil, Assistant Professor at the Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT) in Gandhinagar, took on a difficult project (detecting neonatal abnormalities through spectrographic analysis of four different cry modes) as a solo author.

To Johan, David, Alex, Hemant, and to all the other stellar contributors to *Advances in Speech Recognition*, I offer my wholehearted thanks for your hard work and determination.

A. Neustein

Contents

Part I Mobile Environments

| | |
|--|-----------|
| 1 “Life on-the-Go”: The Role of Speech Technology in Mobile Applications..... | 3 |
| William Meisel | |
| 2 “Striking a Healthy Balance”: Speech Technology in the Mobile Ecosystem..... | 19 |
| Scott Taylor | |
| 3 “Why Tap When You Can Talk?”: Designing Multimodal Interfaces for Mobile Devices that Are Effective, Adaptive and Satisfying to the User..... | 31 |
| Mike Phillips, John Nguyen, and Ali Mischke | |
| 4 “Your Word is my Command”: Google Search by Voice: A Case Study | 61 |
| Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope | |
| 5 “Well Adjusted”: Using Robust and Flexible Speech Recognition Capabilities in Clean to Noisy Mobile Environments..... | 91 |
| Sid-Ahmed Selouani | |

Part II Call Centers

| | |
|--|------------|
| 6 “It’s the Best of All Possible Worlds”: Leveraging Multimodality to Improve Call Center Productivity..... | 115 |
| Matthew Yuschik | |

| | | |
|-------------------------|--|------------|
| 7 | “How am I Doing?”: A New Framework to Effectively Measure the Performance of Automated Customer Care Contact Centers..... | 155 |
| | David Suendermann, Jackson Liscombe, Roberto Pieraccini, and Keelan Evanini | |
| 8 | “Great Expectations”: Making use of Callers’ Experiences from Everyday Life to Design a Satisfying Speech-only Interface for the Call Center..... | 181 |
| | Stephen Springer | |
| 9 | “For Heaven’s Sake, Gimme a Live Person!” Designing Emotion-Detection Customer Care Voice Applications in Automated Call Centers..... | 191 |
| | Alexander Schmitt, Roberto Pieraccini, and Tim Polzehl | |
| 10 | “The Truth is Out There”: Using Advanced Speech Analytics to Learn Why Customers Call Help-line Desks and How Effectively They Are Being Served by the Call Center Agent | 221 |
| | Marsal Gavalda and Jeff Schlueter | |
| Part III Clinics | | |
| 11 | Dr. “Multi-Task”: Using Speech to Build Up Electronic Medical Records While Caring for Patients..... | 247 |
| | John Shagoury | |
| 12 | “Hands Free”: Adapting the Task–Technology-Fit Model and Smart Data to Validate End-User Acceptance of the Voice Activated Medical Tracking Application (VAMTA) in the United States Military | 275 |
| | James A. Rodger and James A. George | |
| 13 | “You’re as Sick as You Sound”: Using Computational Approaches for Modeling Speaker State to Gauge Illness and Recovery..... | 305 |
| | Julia Hirschberg, Anna Hjalmarsson, and Noémie Elhadad | |
| 14 | “Cry Baby”: Using Spectrographic Analysis to Assess Neonatal Health Status from an Infant’s Cry..... | 323 |
| | Hemant A. Patil | |
| | Epilog..... | 349 |
| | About the Author..... | 359 |
| | Index..... | 361 |

Contributors*

Françoise Beaufays, Ph.D.

Research Scientist, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

Doug Beeferman, Ph.D.

Software Engineer, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

Bill Byrne, Ph.D.

Senior Voice Interface Engineer, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

Ciprian Chelba, Ph.D.

Research Scientist, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

Mike Cohen, Ph.D.

Research Scientist, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

Noémie Elhadad, Ph.D.

Assistant Professor, Department of Biomedical Informatics,
Columbia University, 2960 Broadway, New York, NY 10027-6902, USA

Keelan Evanini, Ph.D.

Associate Research Scientist, Educational Testing Service,
Rosedale Road, Princeton, NJ 08541, USA

Marsal Gavalda, Ph.D.

Vice President of Incubation and Principal Language Scientist, Nexidia,
3565 Piedmont Road, NE, Building Two, Suite 400, Atlanta, GA 30305, USA

*The e-mail addresses are posted for the corresponding authors only.

James A. George

Senior Consultant, Sam, Inc., Rockville, MD 1700 Rockville Pike # 400,
Rockville, MD 20852, USA

Julia Hirschberg, Ph.D.

Professor, Department of Computer Science, Columbia University,
2960 Broadway, New York, NY 10027-6902, USA
julia@cs.columbia.edu

Anna Hjalmarsson

Graduate student, KTH, (Royal Institute of Technology),
Kungl Tekniska Högskolan, SE-100 44 STOCKHOLM, Sweden

Maryam Kamvar, Ph.D.

Research Scientist, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

Jackson Liscombe, Ph.D.

Speech Science Engineer, SpeechCycle, Inc.,
26 Broadway, 11th Floor, New York, NY 10004, USA

William Meisel, Ph.D.

Editor, *Speech Strategy News*, President,
TMA Associates, P.O. Box 570308, Tarzana, California 91357-0308
wmeisel@tmaa.com

Ali Mischke

User Experience Manager, Vlingo,
17 Dunster Street, Cambridge, MA 02138-5008, USA

John Nguyen, Ph.D.

Vice President, Product, Vlingo,
17 Dunster Street, Cambridge, MA 02138-5008, USA

Hemant A. Patil, Ph.D.

Assistant Professor, Dhirubhai Ambani Institute of Information and
Communication Technology (DA-IICT), Gandhinagar, Gujarat-382 007, India
hemant_patil@daiict.ac.in

Mike Phillips

Chief Technology Officer, Vlingo,
17 Dunster Street, Cambridge, MA 02138-5008, USA
phillips@vlingo.com

Roberto Pieraccini, Ph.D.

Chief Technology Officer, SpeechCycle, Inc.,
26 Broadway, 11th Floor, New York, NY 10004, USA

Tim Polzehl, MA

Scientific Researcher, Quality and Usability Lab,
Technischen Universität, Deutsche Telekom Laboratories,
Ernst-Reuter-Platz 7, 10587 Berlin, Germany, 030 835358555

James A. Rodger, Ph.D.

Professor, Department of Management Information Systems and Decision Sciences, Indiana University of Pennsylvania, Eberly College of Business & Information Technology, 664 Pratt Drive, Indiana, PA 15705, USA
jrodger@iup.edu

Johan Schalkwyk, MSc

Senior Staff Engineer, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA
johans@google.com

Jeff Schlueter, MA

Vice President of Marketing & Business Development, Nexidia,
3565 Piedmont Road, NE, Building Two, Suite 400, Atlanta, GA 30305, USA
JSchlueter@nexidia.com

Alexander Schmitt, MS

Scientific Researcher, Institute for Information Technology at Ulm University,
Albert-Einstein-Allee 43, 89081 Ulm, Germany
alexander.schmitt@uni-ulm.de

Sid-Ahmed Selouani, Ph.D.

Professor, Information Management Department; Chair of LARIHS
(Research Lab. in Human-System Interaction), Université de Moncton,
Shippagan Campus, New Brunswick, Canada
sid-ahmed.selouani@umcs.ca

John Shagoury, MBA

Executive Vice President of Healthcare & Imaging Division,
Nuance Communications, Inc., 1 Wayside Road, Burlington, MA 01803, USA
Holly.Dewar@nuance.com

Stephen Springer

Senior Director of User Interface Design, Nuance Communications, Inc.,
1 Wayside Road, Burlington, MA 01803, USA
Stephen.Springer@nuance.com

Brian Strope, Ph.D.

Research Scientist, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

David Suendermann, Ph.D.

Principal Speech Scientist, SpeechCycle, Inc.,
26 Broadway, 11th Floor, New York, NY 10004, USA
david@speechcycle.com

Scott Taylor

Vice President, Mobile Marketing and Solutions, Nuance Communications, Inc.,
1 Wayside Road, Burlington, MA 01803, USA
Scott.Taylor@nuance.com

Matthew Yuschik, Ph.D.

Senior User Experience Specialist (Multichannel Self Care Solutions),
Relationship Technology Management, Convergys Corporation,
201 East Fourth Street, Cincinnati, Ohio 45202, USA
yuschikholmes@comcast.net

Part I

Mobile Environments

Chapter 1

“Life on-the-Go”: The Role of Speech Technology in Mobile Applications

William Meisel

Abstract The mobile phone is becoming an indispensable personal communication assistant and multifunctional device; increasing electronic options in automobiles and other mobile settings extend this “always-available” paradigm. The range of applications creates user interaction issues that can’t be fully solved by extending the graphical user interface and keyboard to these small devices. Speech recognition, text-to-speech synthesis, and other speech technologies are part of the solution, particularly since, unlike PCs, every mobile phone has a microphone and speech output. Two supporting trends are today’s speech technology’s demonstrable ability to handle difficult interactions, e.g., the free directory assistance services, and a resulting interest by deep-pocketed large firms in using and promoting the technology and its applications. This chapter digs deeper into these points and their implications, and concludes with a discussion of what characteristics will make voice interaction an effective alternative on mobile devices.

Keywords Mobile device • Smartphone • Speech recognition • Text-to-speech synthesis • Mobile Internet • Overburdened graphical user interface • Mobile search • Mobile user experience • Speech interface

1.1 Introduction

The mobile phone is one of those seminal developments in technology upon which other technology innovations are built. Consider the ways it represents a new paradigm:

- The mobile phone lets us be connected wherever we are. Both data and voice channels let us be connected to services as well as people.

W. Meisel (✉)

Editor, Speech Strategy News, President, TMA Associates, P.O. Box 570308, Tarzana, California 91357-0308,
e-mail: wmeisel@tmaa.com

- It is truly a “personal telephone” – even more personal than a “personal computer,” which is often shared. As a personal device, features and services can be tailored to our individual preferences and needs.
- It is becoming an extension of ourselves since we can always have it with us. It, for example, makes it unnecessary to remember phone numbers. In the future, we are likely to think of it as our friendly personal assistant, providing a host of services.
- It represents part of an explosion of communication options, and “smartphones” can support all of those options – voice, text, email, and more.
- It tends to be both a business and a personal device – we don’t carry two mobile phones.
- Unlimited calling and/or data plans with a fixed monthly fee make the incremental cost of one more call or data access “free.” We used to think of telephone calls as costly; now they are much the same as Internet access on a PC, which we think of as free because we pay a fixed monthly fee.

An automobile is also a mobile device – it won’t fit in your pocket, but it even has “mobile” in its name. Increasingly, automobiles have complex built-in electronics such as navigation systems, and many have connectivity to your other mobile devices, such as mobile phones and music players. With all these options, and the obvious safety issues, controlling these devices while driving becomes a challenge.

There are other mobile systems that are increasingly complex, some of which are not so well known. For example, in warehouses, when workers travel around to bins or shelves – often on electric vehicles or forklifts – picking merchandise for an order, they receive orders through a wireless system. Their hands and eyes are occupied, but they need to hear what needs to be picked up and tell the warehouse management software when an order has been picked up or if an item is out-of-stock. The issue of hands-free communication is similar to that of automobiles. A March 2009 report from Datamonitor assessed the 2008 global market for voice systems in warehouses at \$462 million.

There are many other examples of enterprise mobile applications. Another growing market is in healthcare, with workers whose hands are often gloved or occupied with patient care, but need to document procedures. In most enterprises, employees increasingly must deal with multiple sources of messages from wherever they are, including when they are out of the office. “Unified Communications” solutions that use speech technology help make this easier, e.g., by allowing emails to be read as speech over the phone using text-to-speech synthesis or by delivering voicemails as text.

The incorporation of speech technology in the user interface will be accelerated by the growth of features in mobile systems. The use of speech recognition in particular will grow rapidly, but text-to-speech synthesis and speaker authentication will also prove important.

1.2 The Need for Speech Recognition on Mobile Devices

Let’s examine the motivations for speech recognition in particular.

1.2.1 An Overburdened Graphical User Interface

The graphical user interface (GUI) on PCs has fueled its usability and growth. Most smartphones in late 2009 attempted to transplant the GUI concept to mobile phones with minimal innovation. Touch, for example, was added as an alternative pointing device, adding support for multi-finger gestures to zoom in or out and for other functionality. The transplantation was effective in giving users something familiar they could use without a user’s manual, but using the GUI with a small screen and inadequate keyboard was not an easy process.

One could argue that the GUI has even become overburdened on PCs, with too many applications, too many files, and too many features in each application. The problem is compounded on mobile devices.

The more than one hundred thousand downloadable applications for the iPhone are symptoms of a failure of the basic user interface. The number of applications for PCs probably hasn’t reached such numbers after more than a decade of use – the large screen and convenient keyboard and mouse make the basic operating system sufficient to handle most functions within general applications or web browsers. The mobile applications are necessary to make certain operations more easily usable.

Speech can provide a significant, perhaps critical, alternative for the user interface, and the rapid proliferation of voice control, voice search, and voice dictation options for mobile phones suggests that this is recognized by many vendors. At the end of 2009, however, speech was an “add-on” that required applications to work around the limitations of the mobile operating systems, rather than being seamlessly integrated with other user interface options. Mike Phillips, co-founder and CTO of Vlingo, which provides one such voice option for mobile phones, noted in an interview in *Speech Strategy News* in December 2009: “Not only does the speech functionality need to be integrated in a way that a user can speak into any application, but if it is truly part of the operating system, then application designers can start to take into account the fact that users can speak to their applications and may make some different design decisions to better optimize their applications for this use case.”

1.2.2 The Need for a Hands-free Option While Driving

“Distracted driving” has attracted the attention of lawmakers and regulatory agencies. The issue is in part the misuse of mobile phones while driving, e.g., dialing or texting. It is unlikely that the use of mobile phones while driving can be successfully forbidden, even if legislators were willing to enact such an unpopular law, since hands-free use can be essentially impossible to detect from outside the vehicle. Further, lawmakers would logically have to outlaw talking to passengers, since it’s likely that that is equally distracting. Thus, using speech recognition to allow hands-free control of communications devices is, practically speaking, a required option for mobile phone makers and automobile manufacturers. Control of music systems, navigation systems, and the increasing number of electronic

options within vehicles also motivates a speech interface, both for hands-free use and to avoid confusion with multiple buttons and knobs.

1.2.3 Lack of Uniformity

Personal computers evolved such that one or two operating systems and suites of applications quickly came to dominate. One can usually sit down at any PC and use basic functions. Each wireless phone – and often each wireless provider – offers a significantly different experience. A voice interface can introduce an intuitive, consistent option across many devices.

1.2.4 Advanced Features for Basic Phones

Speech in the network can add features for phones with no data channel (other than texting). While smartphones seem to attract the most attention, the majority of phones today and for a long time are basic voice devices. The apparently unnoticed implications – the *voice channel* may grow to be an important way for those users to access some of the services now accessed by the data channel on smartphones.

Smartphones will grow to comprise roughly 60% of new handsets sold in the U.S. by 2014, according to a forecast in late 2009 from Pyramid Research.¹ The forecast found that smartphones will represent 31% of new handsets sold in the U.S. in 2009, more than double from 15% two years prior. Infonetics Research² also released a report in late 2009, estimating that smartphones would post a 14.5% increase in the number of units sold worldwide in 2009. While this growth is impressive, the forecast shows that basic phones will remain in the majority for a long time.

Research firm Gartner, Inc. estimated in late 2009 that about 309 million handsets were sold in the third quarter, up 0.1% from a year earlier.³ Sales of smartphones increased 12.8% to reach over 41 million units. Again, the growth rate of smartphones suggests their continuing appeal, but their sales by these estimates were only 13% of the total. One way of viewing these results is thus that “there were 309 million more prospects added to voice-channel services (since smartphones have a voice channel), and 41 million more to data-channel prospects.” There is no web surfing on basic phones.

Voice services reached from any phone today include the free directory assistance services, some of which provide much more than just local business connections. Those services may grow into major general “voice sites” supported largely by advertising.

¹ *Smartphone Forecast: Operator Strategies Will Fuel Growth in Emerging Markets*, December 2009, Pyramid Research (<http://www.pyramidresearch.com>).

² *Mobile/WiFi Phones and Subscribers*, November 2009, Infonetics Research (<http://www.infonetics.com>).

³ “Forecast Analysis: Mobile Devices, Worldwide, 2003–2013, 4Q09 Update,” December 2009, Gartner, Inc. (<http://www.gartner.com>).

1.2.5 The Availability of a Non-speech Option

Although it may seem contradictory, the availability of modes of user interaction to accomplish a task without using speech can make a speech interface more acceptable. One can't talk in every environment, so the ability to accomplish a task, even if less efficiently than with voice, encourages the incorporation in the device and the network of many applications and features.

In addition, some information can best be delivered by means other than speech, even if it is retrieved by speech commands. On mobile phones, displaying options, text, or graphics (such as maps) is an option. Even on a voice-channel call, the ability to deliver some information as email or a text message increases the usability of the speech interface.

1.2.6 Making Voice Messages more Flexible

Voice mail is a necessity for telephone calls, but it is less convenient than text messages or email, which can be reviewed at leisure, dealt with out of order, and can be easily stored and in some cases easily searched. “Visual voicemail” is a growing application, allowing message headings to be displayed as a list and listened to out of order.

Converting voicemail to text using speech recognition makes visual voicemail considerably more useful. Since voice mail has been around a long time, why are we seeing voicemail-to-text services proliferate now? In part, it's because handling many and long voicemails is more difficult on mobile phones than desktop phones, where it is typically easier to take notes on the voice messages.

1.2.7 Open-source Wireless Phone Platforms that Support Speech Technology

Google's Android open-source mobile phone operating system is available under a liberal license that allows developers to use and modify the code, offered through a group Google initiated, the Open Handset Alliance. The Open Handset Alliance has at least basic speech recognition available as part of the open-source package, making it easier for independent developers to economically include a speech option in their software.

1.2.8 The Dropping Cost of Speech Technology

Speech technology licenses, as with most technology solutions, are likely to get cheaper as volume expands, making the speech automation discussed as part of other trends more affordable. Barriers caused by the cost of speech technology

licenses should diminish. And the cost of computing power to run the speech recognition software continues to drop.

1.2.9 A Desire to Provide Web Services on Mobile Phones

The Web has created many successful businesses, and companies want to replicate that success on an increasingly important platform – the wireless phone. The data channel makes this possible, but in many cases, such as while driving, it is difficult to use Web services without a speech option. Further, many Web sites have not been adapted to mobile phones, and are difficult to navigate on small screens. Speech interaction may be one way to deliver services equivalent to what a visual web site delivers.

The desire to deliver Web services on mobile phones was emphasized in a keynote address at a conference in 2009 delivered by Marc Davis, Chief Scientist, Yahoo! Connected Life: “Speaking to the Web, the World, and Each Other: The Future of Voice and the Mobile Internet.” He noted that mobile is a unique medium with tremendous opportunities in terms of scale, technological capabilities, and how it integrates into people’s daily lives. The mobile search use case is different from how consumers use search on the PC, and speech is a natural input method for search on mobile devices. Davis said that voice-enabled mobile Internet services will enable people to interact with the Web, the world, and each other and will change the role of voice as a medium for search, navigation, and communication. Davis emphasized the role of use context – where, when, who, and what – to make intelligent interpretations of a user query.

The investment firm Morgan Stanley issued a *Mobile Internet Report* near the end of 2009, a 694-page report that in effect declared Mobile Internet Computing as the technology driver of the next decade, characterizing the 1990s as being driven by Desktop Internet Computing, the 1980s by Personal Computing, the 1970s by Mini Computing, and the 1960s by Mainframe Computing.⁴

To outline a few points from the report:

- *Market impact of smartphones isn’t full measured by market penetration;* mobile Internet usage reflects the usability of the smartphone. For example, Morgan Stanley says the Apple iPhone and iPod Touch are responsible for 65% of mobile Internet usage, although they represent only 17% of global Smartphones.⁵ This implies that usability is a key factor, but the long presentation makes no mention of speech recognition as a factor in Mobile Internet growth.
- *Material wealth creation and destruction should surpass earlier computing cycles.* The report notes that winners in each computing cycle often create more market capitalization than in the last, and that past winners often falter.

⁴The full report is available from Morgan Stanley: http://www.morganstanley.com/institutional/techresearch/mobile_internet_report122009.html

⁵Perhaps these numbers discount mobile email use as part of the Mobile Internet, since Research In Motion’s Blackberry is particularly popular for this feature, and continues to show strong growth.

- *The Mobile Internet is ramping up faster than the Desktop Internet did.* Morgan Stanley believes more users may connect to the Internet via mobile devices than through desktop PCs within 5 years.
- *Five key factors provide the foundation of growth:* 3G adoption, social networking, video, VoIP, and “impressive mobile devices.”
- *“Massive mobile data growth” will drive the market.* The focus of the report is on the data channels impact, as opposed to the voice channel.
- *In emerging markets,* mobile may be the primary means of access to the Internet.
- *Mobile phones are moving from a focus on voice communication to multipurpose devices.* One chart in the report shows that the average American cell phone user spends 40 min a day on a mobile phone, making calls 70% of that time. The average iPhone user, by contrast, spends 60 min on the device but makes calls only 45% of the time. The rest of those 60 min are spent texting, emailing, listening to music, playing games, and surfing the Web.

But what is the “Mobile Internet”? The report seems to emphasize the “Internet” in that phrase, treating the Mobile Internet as the Web accessed by a mobile device. I would emphasize the “Mobile” in the phrase as the key to growth. Our mobile device will almost certainly have a wireless connection, so it keeps us connected to others and to information sources; it will even tell us where we are and how to get somewhere else. We can take a mobile device everywhere, and, since it can always be with us, we can come to depend on it. (Most people have experienced the panic that rises when a mobile phone is misplaced or lost.) Since we *must* have our primary mobile device with us (almost certainly a mobile phone), that device will tend to increase the number of functions it provides, at the expense of other mobile devices such as audio players.⁶

To repeat a theme previously mentioned, all this functionality on a small device strains its usability. Touch screens help, but, unless we evolve smaller fingers to adapt to the device, there are limitations to touch technology.

An adequate user interface will allow the natural growth of mobile devices as Morgan Stanley anticipates, but it isn’t the Internet per se that creates that growth. The Web is a well-established phenomenon of the last decade, as the report itself points out. It is mobility and making that mobility feasible that marks the trend.

Another conundrum raised by the report is the convenient description of one decade-long predominant trend in computing after another. Many areas of technology clearly grow exponentially, Moore’s law of chip complexity being one of the more famous. The economist W. Brian Arthur, in his 2009 book *The Nature of Technology: What It Is and How It Evolves*, goes into great depth on how technologies evolve and why technology growth accelerates. In part, it is because technologies are assembled from other technologies; and, as the toolkit of available technologies grows, invention becomes easier. Yet, the apparent linear progress of computing breakthroughs belies this supposed acceleration.

⁶A potential hurdle is battery limitations, but I suspect this will be overcome in the long run by easily used induction chargers in coffee shops, in autos, and other places we frequent, chargers that don’t require a physical connection.

Perhaps the mystery lies within us – us humans, that is. A technology must be *used* to be of value, of course. The trends that the Morgan Stanley report cites are trends in *human use of computing*. Most humans I know don't change their habits exponentially; most are in fact a bit resistant to change. It takes exponential improvement in usability to persuade humans to move (even linearly) toward adoption of a technology that requires human use.

Smartphones depend on the understanding of the GUI on PCs and the keypad on all mobile phones to make them acceptable to their owners. Most innovations are clever adaptations to a small device, rather than breakthroughs.

One could argue that the prediction of the importance of the Mobile Internet over the next decade requires that we overcome this resistance to change with a true and effective innovation. Fortunately, all mobile phones have a microphone.

1.3 The Personal Telephone

Another aspect of mobile phones is that they are *personal* devices. We have computers, and we have *personal* computers – PCs. We have telephones, and we have *personal* telephones. Unlike telephones in homes and businesses, wireless phones are almost always associated with one individual. And, unlike those tethered devices, it is almost always *with* that individual.

These simple facts are not a simple development. The personal phone is a fundamental paradigm shift. There are a number of components to this shift that can make a voice interface more effective:

1. *Personalization*: A wireless phone identifies itself and implicitly identifies you when it places a call. If a caller elects to use a service that employs personalization, the service can remember preferences and tendencies from call to call. Speech recognition can be more accurate when one can bias it toward previous choices of the individual and even their specific voice and pronunciation. Dialogs can be more compact if the user has indicated preferences by specific acts.
2. *Localization*: When a device has GPS capability, it can indicate where you are; and localization is a powerful tool, particularly for advertising. Location information can avoid the need to speak that information.
3. *Availability*: The device is always with its owner, making services and features always available. This will increase dependence on the device; for example, few people memorize telephone numbers any longer, since they are in their mobile phone contact list. This motivates becoming familiar with useful services, and can make the device central to the owner's activities. The device is a constant companion, and a voice interface can humanize that companion and create the mental model of a friendly personal assistant. (Over-personalization, giving the assistant a perhaps annoying "personality," does not have to be part of the experience.)
4. *Retention and access to information*: Because the device is always available, you may also want to be able to do things with it that you would otherwise do on a PC. You may want key information that you normally access by PC available to you on-the-go. That drives features such as access to email and the maintenance of a digital contact list.

5. *Long personal lists*: Accessing a list of information, such as contacts or songs, is a perfect fit for speech recognition. No “menu” is necessary and the user knows what to say. In such applications, a personally created list is available as text and can be automatically converted into a speech grammar without effort by the user.
6. *Multifunctionality*: The portable nature of the device motivates other functions that the owner would like available when mobile. A camera, music player, navigation device – why carry multiple devices if one can do it all? The large number of options makes a voice interface for finding features increasingly attractive.

These trends demand a voice-interactive “personal assistant” model in the long run. Perhaps we should resurrect the concept of the Personal Digital Assistant (PDA) with a new paradigm.

1.4 A Paradigm Shift in the Economics of Phone Calls

As this book was being compiled, mobile service providers were providing both prepaid and conventional plans that made it economical for subscribers to effectively have unlimited minutes for voice calls. Since there are only so many minutes one can be on a voice call in a day, the service providers can be sure that the minutes are in fact limited. While service providers were concerned over high usage of data channels because of high-demand tasks such as downloading video, unlimited plans also often included unlimited data.

For consumers with unlimited plans, the cost of one more phone call is perceptually zero, and the length of calls doesn’t matter. That is a paradigm shift from historical perspectives on phone calls as a costly means of communication that had to be kept short. Anyone observing a teenager using a mobile phone to talk to friends probably feels that the younger generation thinks calls are free already.

To understand how that paradigm shift may affect voice usage, consider email and Web access. They are perceived as free, although customers do pay a monthly fee for unlimited Internet access, analogous to unlimited calling plans. Isn’t it likely that eventually telephone calls will be accorded the similar perception that calls are “free”?

VoIP calls use the data channel and thus are part of the data plan. If VoIP usage increases, it will be hard to continue to make a distinction between voice and data on a cost basis.

As the paradigm shift toward free or low-cost telephony develops, it could have implications for automated phone services, including those using speech technology:

- *Stay on the line, please*: Customer service lines could increasingly adopt a philosophy that, once a customer’s initial reason for calling is resolved, the service should encourage continued interaction to inform the customer about other options or the company’s offerings in general (“upselling” or “cross-selling” being examples). Customers could be offered outbound alerts on the availability of some upcoming product or reminders relevant to the company’s offerings. The longer the call (outbound or inbound), the more motivation to automate it, since the cost of a call

involving an agent is proportional to the length of the call. Agent interaction should be preserved for the cases where it is most needed.

- *Call me for fun:* Some telephone “services” could be ones that customers call for entertainment, a practice certainly common in web surfing. Some calls of this genre will be motivated by conventional advertising. These services could be made unique by making them interactive, as opposed to passive listening, so that callers can call the same number often, yet have a different experience each time. Part of this “conversational marketing” could be funded from the company’s advertising/marketing budget, and conventional creative talent could become involved in designing the interaction. Advertising budgets for most companies easily exceed budgets for call centers by orders of magnitude.

1.5 Humans in the Loop

Some voicemail-to-text and voice notes services use human editors to correct speech recognition errors before sending the transcription to the end user. (This is the most common case in medical dictation, some of which is done over internal phone systems.) Using editors of course increases accuracy. Speech recognition, even with editors, can reduce costs compared to transcribing speech without pre-processing by speech recognition. A typical estimate is a 50% reduction in human transcribers’ time.

One role for using editors is that the corrections can be used to improve the accuracy of speech recognition if used to fine-tune the speech recognition parameters. Such adaptive speech recognition has long been incorporated in some dictation systems, including medical systems.

Review and adjustment of speech recognition using people occurs in call centers as well, although in a less obvious way. Automated customer service applications require tuning by review of what callers unexpectedly say that causes failure of the automated system. Dialog-design experts have long used recordings of call center conversations and similar tools to adjust the speech recognition grammars to cover those cases. Speech analytics – speech recognition used with software to find problem calls – can help this process.

Adaptation can make a speech system get smarter over time. In some of the more difficult speech applications, editors can initially improve system acceptance and in the long term reduce the need for that human assistance.

1.6 Other Speech Technologies and Mobile Applications

1.6.1 Text-to-speech

We’ve emphasized speech recognition in this discussion. Text-to-speech synthesis is part of many of the applications we’ve discussed. It allows, for example, text information in databases to be spoken to a caller without the need for that information to have been previously reported.

Today’s text-to-speech largely uses concatenated small slices of recorded speech and sounds very natural. It is usually recognizable as synthetic because of occasional mispronunciations and misplaced stress, but it is easily intelligible.

1.6.2 Speaker Authentication

Speaker authentication – sometimes called “speaker verification” or “speaker identification” – has not achieved the penetration it deserves. Part of the hurdle to its use is that it requires an enrollment by its very nature. It is increasingly used in call centers to shorten authentication that might otherwise require a number of security questions and the use of an agent. For mobile phones, however, a key use might be on the device itself to make sure the device can’t be used without the owner’s permission. This might become more important as these devices include an increasing amount of potentially private information, such as contact lists and emails.

1.6.3 Audio Search

“Audio search” uses speech recognition to find specific search terms within unstructured audio, including the audio track of video. It allows, for example, general web search to include the content of audio/video files, not just their metadata. This technology is not particular to mobile uses, but can form part of the “Mobile Internet” experience.

1.7 What’s Required to Optimize the Mobile User Experience?

This chapter has discussed the motivation for and general functionality of speech interfaces on mobile devices. Let’s dig deeper into what will make those speech interfaces effective.

1.7.1 Just Say What You Want

The simplest user manual is “just say what you want” (SWYW). This is of course the ideal, and is difficult to achieve, both in terms of full generality of the speech recognition and the limitations of integration with mobile operating systems and applications as they are at this writing.

If one could in fact achieve this flexibility, the speech interface could become dominant as the model of the user interface on mobile phones. This might seem unlikely, since one can’t always use speech; there are many situations where silent interaction with the mobile device is required. The key is that a user might view non-speech interaction with the device as “type what you would say.”

This approach perhaps appears to require speech recognition technology that is overly difficult to implement effectively. On the surface, it would appear to require deep natural language understanding, which I believe to be too high a hurdle for today's technology. In fact, if the command were truly unconstrained, then perhaps SWYW is too ambitious because the user's query in such cases would cut across too large a subject domain to be properly understood and acted upon by the mobile device or network services supporting the device.

However, I don't believe a mobile phone user's request will be unconstrained. The implicit instruction is to say what they want the mobile phone or a mobile service to deliver. One doesn't walk into a pizza parlor and say to the clerk taking the order, "Is my prescription ready?" Similarly, the implicit constraint is what a mobile phone or service can do. Further, a system interpreting the statement can take advantage of the personal nature of the mobile phone to have context on the user, among other things, where they are, who their contacts are, and what they have said before.

Further, SWYW has a built-in constraint. It is implicitly a command. One wouldn't start dictating an email or text message in response to "say what you want," but would more likely say "dictate a message" or "send an email to Joe" first. Today's systems require such a hint. (More on dictation in a later section.)

In a voice user interface, there will be categories of functionality such as navigating to an application on the phone, connecting to a network-based service, dialing a contact, conducting a web search, dictating a text message, etc. The user can learn quickly that keywords such as "search," "dial," or "dictate a message" will make the result more reliable, and the system's job in interpreting at least the general context of the message is limited to the domains it can handle. It can do the equivalent of saying "I don't understand" (e.g., a beep) if it can't categorize the request into one of these domains. Such feedback will help the user learn what works. One expects that a user could quickly learn to provide some context for a command when necessary, as long as the specific way that the context was provided was flexible and intuitive.

There is another reason to believe that SWYW isn't too high a hurdle. The system, like a pizza clerk, knows the limits of what can be delivered. If you say "pepperoni," the clerk will understand "pepperoni pizza" and ask "what size?" If you say, "size ten," you will get a stare of incomprehension. The computer can say the equivalent of "what?" Humans use context to understand, and machines must also do so.

1.7.2 Talking to the Text Box

One model of voice user interface today on mobile phones is being able to dictate into any text box, or, in some cases, into any part of an application that allows entering text. Some voice applications also have their own text box that appears when the application is launched, perhaps by clicking on a button on the phone or

an icon on the screen. The voice application then has the option of interpreting the voice command to launch an application appropriate to the command, rather than requiring manual navigation to that application.

Another model available is dictating into a “clipboard” application that is part of the phone’s operating system. The contents of the clipboard can be pasted into most applications.

One deficit of these approaches as they have been implemented to date is a lack of interactivity. Once one says something into a text box, some action is usually taken that drops one out of the speech interface, e.g., displaying a list of search results. There is no dialog. Dialog is a powerful way to resolve ambiguities. For example, in web search, there is often a number of ways to interpret a search request, and a long results list with many non-responsive options is more difficult to view on a mobile phone. Ideally, these interfaces will evolve to allow more interactive dialog when it can be useful.

1.7.3 Dictation

While a command/request to a mobile phone may be implicitly constrained, dictation of emails, text messages, or voice notes is essentially unconstrained. Dictation of free-form text is an option supported by a number of companies, usually with a small client application in the mobile device and speech recognition within the network. This approach uses the data channel, and the quality of speech that can be delivered over the data channel is usually better than the voice channel.

A key difference in dictation and voice requests is that the dictation is intended to be read and understood by a human, not a computer. The composer can also edit the message before sending it. A voice command must, in contrast, be understood by the computer. Thus, the task for dictation is different. It is harder in some ways and easier in others than a “say what you want” user interface.

The mobile phone being a personal device eases the dictation task. Most dictation systems tune both to the vocabulary usage and voice of the user. One dictation application for mobile phones downloads and incorporates contact names (if the owner allows) and can thus be accurate in transcribing proper names that are in the contact database.

In the case of voicemail-to-text services, an area growing rapidly, there is more context than one might initially think, because in part it is a personal voicemail inbox. As an example, if you are Joe, I am likely to say, “Hi, Joe, this is Bill.” I am unlikely to say “Hi, Sally” unless I’ve dialed a wrong number. And, if I don’t leave a last name, I probably am someone who calls you often, so that that “Bill” will be more likely than “Phil” if you don’t have a frequent contact named “Phil.” And I’m pretty likely to end the call with something like “Bye.” This is context specific to voice mail, and unlikely in, for example, a medical report.

1.7.4 Multi-modal User Interfaces

A voice user interface should be viewed in the context of the evolution of user interfaces. It can be part of addressing the growing complexity of our use of the Web and PC applications, as well as multifunction mobile devices and entertainment systems. Perhaps the term “voice user interface” is misleading; the appropriate approach is to make voice a complement to other modalities available, not a complete replacement. The GUI didn’t drop the keyboard as an option when it added a pointing device; and Web and PC search models didn’t replace the GUI as an interface.

Viewing the voice user interface as an evolution that enhances the prior generation of user interface innovations, rather than replacing them, is a useful approach. Certainly, when a hands- and eyes-free interaction is desired with an otherwise GUI- or text-oriented interface – for safety or other reasons – pure voice interaction provides an option. But even in this case, information can be delivered as text for later use. The issue is what best serves the user.

1.7.5 Universality

Ideally, a speech interface should be consistent across applications and platforms. Consistency has been key in driving the acceptance of GUI interfaces; pointing and menu selection, for example, is a familiar process despite many different details. Today, that consistency is lacking for voice interfaces. It is one experience to call to get directory assistance, for example, and quite another to call a contact center and be presented with a menu, and yet another to dictate a text message.

At the time of this writing, when the average person is asked about their interaction with a voice interface, they mention call centers. To the degree there is uniformity in call center speech interactions, it is a “directed-dialog” model, where the caller is told what they can say at every step. While this is a style of interaction that a customer might come to expect, it differs with each company in its content and style. There isn’t much that is intuitive about most of these interactions.

Can we establish and build on a baseline to make the voice user interface in applications as diverse as mobile phones and call centers as familiar and acceptable as today’s GUI? What could that baseline be?

A speech-recognition baseline should:

- Be intuitive so that no user manual is required;
- Translate from one platform to another, so that one can move to a platform not used before and have a basic understanding of how to use the speech interface;
- Form the basis for understanding extensions of that baseline that may involve variations on the speech interaction; and
- Take advantage of other modalities as fallback when speech isn’t an option, ideally maintaining the same mental model as the speech interface.

This chapter suggested that one possible user mental model for a mobile phone interface is “say what you want.” The alternative for maintaining the same mental

model when one can't talk, as noted, is “type what you would say.” If dialog to clarify requests is added to these, the user interface might be general enough to be considered universal.

One complication in creating a consistent voice user interface experience is that mobile phones have two distinct ways of connecting with computers or people. One is the conventional voice channel, and the other the data channel. The data channel supports multimodality more easily, since it can display, for example, a list of options in response to a voice request. The voice channel can deliver some information as text by email or text message if properly set up, but this is hardly interactive. Switching from a voice interface on one “voice site” by a phone call to a voice interface on another can result today in a much different experience.

Automated directory assistance services over the voice channel can be reached from any phone, and are becoming widely used. Some already offer, on the same call, weather, driving directions, stock quotes, movie times, jokes, and remember your home address if you provide it. Perhaps the way to maintain a consistent experience is to stay within that voice site, a site designed to be consistent across functions. Perhaps if someone does this really well, they will dominate the voice site “business” for voice-only calls. As noted, the voice channel will continue to be the only channel available to a majority of mobile phone subscribers for many years, particularly if international users are included.

1.8 Mobile Workers

We previously noted that mobile workers, e.g., in warehouses and healthcare, can make use of speech interaction and wireless networks to increase efficiency and avoid errors. One example is the products of Vocollect, Inc., one of a number of vendors addressing this market. Every day in 2009, Vocollect helped over 250,000 workers worldwide to distribute more than \$2 billion dollars' worth of goods from distribution centers and warehouses to customer locations.⁷

One example is the parts center of IHI Construction Machinery Limited, a Japanese company that manufactures and markets large-scale construction equipment including mini-excavators, hydraulic shovels and cranes, and associated environment-related equipment. Vocollect Voice is used by IHI for cycle-counting, receiving inspection, storage, picking, and shipping inspection, supporting parts control for approximately 60,000 items of varying sizes at its parts center in Yokohama. Before introducing the voice solution, the IHI parts control center used hand-held terminals or paper labels.

The company has achieved a 70% reduction in work errors from its 1-year implementation of Vocollect Voice, helping the company attain a 99.993% operating accuracy. The company also realized a 46% average improvement in productivity, reducing the number of workers per shift by 50%.

⁷“Vocollect continues expansion into Asia,” press release, October 2009, Vocollect, Inc. (<http://www.vocollect.com>).

1.9 Conclusion

Speech recognition is a technology, of course, not a product in itself. The mobile phone, however, has given it the perfect platform to demonstrate that it has matured as a technology to the point where it can support powerful applications, and, arguably, do most of the heavy lifting in a user interface. Historically, expectations that the technology could match human listening, speaking, and understanding skills have hampered acceptance when it didn't jump that high hurdle. If users instead compare it to a frustrating experience trying to use a graphical interface on a small device, that barrier will be lowered. In the last analysis, any user interface can be designed well or poorly, irrespective of the technology. This section of the book contains in part perspectives on what works optimally and what doesn't perform as well in the mobile environment.

Chapter 2

“Striking a Healthy Balance”: Speech Technology in the Mobile Ecosystem

Scott Taylor

Abstract Mobile speech technology has experienced an explosion of adoption across a variety of markets – from handsets and automobiles to a variety of consumer electronic devices and even the mobile enterprise. However, we are just scratching the surface on the benefits that speech can provide not only consumers, but also carriers and manufacturers. This chapter takes a closer look at the advent of speech technology in the mobile ecosystem – where it is been, where we are today, and where we are headed – keeping in mind the delicate balancing of a variety of multimodal capabilities so as to optimally fit the user’s needs at any given time. There is no doubt that speech technologies will continue to evolve and provide a richer user experience, enabling consumers to leverage the input and output methods that are best suited for them moment to moment. However, the key to success of these technologies will be thoughtful integration of these core technologies into mobile device platforms and operating systems, to enable creative and consistent use of these technologies within mobile applications. For this reason, we approach speech capabilities on mobile devices not as a single entity but rather as part of an entire mobile ecosystem that must strive to maintain homeostasis.

Keywords Mobile ecosystem • Multimodal navigation • Multimodal service calls • User experience • Speech technologies • Integration into mobile device platforms and operating systems • User interface challenges to designers • Hands-free • Enterprise applications and customer service

S. Taylor (✉)

Vice President, Mobile Marketing and Solutions, Nuance Communications, Inc.,
1 Wayside Road, Burlington, MA 01803, USA
e-mail: Scott.Taylor@nuance.com

2.1 Introduction

The availability of computing power and network connectivity in automobiles, mobile phones, and other mobile devices has led to an explosion of available applications and services for consumers. Maps and navigation, the advent of social networking sites like Twitter and Facebook, email, web search, games, and music and video content have become commonplace on mobile devices, and are now emerging as services available in cars and in other electronic devices.

But as these new services and applications become more popular, they pose many user interface challenges to designers. For instance, devices are limited in computing power, display size, and the keyboard is small and difficult for many people to use. Also, the convenience of mobility creates situations where the users are not always able to keep their eyes and hands on the device...walking, engaging in conversation, working out at the health club, and the obvious – driving a car. With these challenges in mind, device manufacturers have invested heavily in technologies that ultimately improve the user interface experience, such as predictive text, touchscreens, and speech technology.

Speech technologies, including both speech recognition and text-to-speech, have been popular for use in mobile applications for decades. However, until recently, that popularity was limited to niche applications, such as voice-dialing or assistive applications, to help the disabled use mobile technology. In the last few years, there has been a rapid expansion in the breadth of mobile applications, leading to an increased demand for speech technology.

Historically, speech technologies have been preloaded on devices at the factory...on mobile phones, in automotive in-car platforms, or in gaming devices. It is available in the device right off the shelf. However, as third generation (3G) and fourth generation (4G) wireless data networks have become prevalent and more robust, application providers are now using the additional computing power that is available to provide more advanced speech capabilities to mobile devices and downloadable applications.

Today, mobile speech applications tend to be focused on core services such as navigation, dialing, messaging, and search. In the future, we will see speech used in a wide variety of mobile applications, including entertainment, social networking, enterprise workforce, mobile banking and payment, customer service, and other areas. Speech technology will also become available in a wide array of devices.

2.2 The First Mobile Voice Applications

2.2.1 *Voice Dialing and Voice Commands on Phones*

One of the first mobile voice applications to emerge in the 1990s was voice dialing, which allowed users to press a button and speak a number or name to call so that the user could place phone calls without looking at the keypad and trying to find numbers.

Initial voice-dialing technology used speaker-dependent technology, or “voice tags.” With speaker-dependent technology, the user of a device needed to go through an enrollment process, where they would speak recordings of the digits and names that would be used for dialing. Each digit or name typically had to be recorded one to three times, and the voice-dialing application would only work for the user who enrolled.

One advantage of speaker-dependent dialing is that it was language independent. A device equipped with “voice tag” capabilities could be used by a speaker in any language. However, the voice-dialing applications could not automatically learn new names as they were added to the contact list, and the enrollment process was frustrating for many users. Unfortunately, many users formed early and negative impressions of mobile speech recognition capabilities from these early speaker-dependent systems. Today, as the technology evolves, it continues to be a challenge for the industry to overcome those negative first impressions.

Computing power and memory footprint continued to increase on mobile devices. Device manufacturers soon added more sophisticated phonetic speech recognition capabilities to the device. Phonetic speech recognition used acoustic speech recognition models trained on a wide variety of speaker voices and styles, and recognized phonemes rather than word templates, and had the following advantages:

- No user enrollment or training was required.
- New words and contact names could be added dynamically. If a new contact name is added to the contact list, then it could be recognized by the voice dialer using standard phonetic pronunciation rules.
- The application could recognize flexible manners of speaking. For example, a user could say “Call John on his mobile,” or “Dial John Smith on his cell phone.” If the application was correctly programmed, it could handle a great deal of flexibility.

Some voice command applications could also be programmed to recognize a long list of commands, beyond just dialing. In fact, some phones today can recognize 50–100 voice commands to control the device. Popular hands-free commands include:

- “turn Bluetooth on/off”
- “send text to <contact-name>”
- “check battery”
- “check signal”
- “go to camera”
- and more

Unlike voice tags, phonetic speech recognition on the phone was not speaker dependent, but rather language dependent, meaning that the software works out of the box for any user, but only recognizes specific languages and dialects. With that in mind, it became very important for on-device technology to support many languages given today’s global landscape. And while this language-dependent technology can support a variety of commands and speaking styles, it nevertheless requires the user to use gate commands like “dial Jim” or “go to voicemail.” In this instance, users must have a sense of which commands are supported on the device – potentially creating an additional learning curve for some users.

2.2.2 The Advent of the Hands-free Experience on the Phone

Voice dialing and other voice commands were expected to work well in situations, where the user's hands and eyes were not completely free, and so it was important that these applications provided a minimal attention interface.

Implementers of speech recognition systems on a device needed to consider the amount of button pressing and holding required to use speech recognition. The simplest and safest interfaces required only a simple button push, as described in the following sequence:

- User pushes a button and quickly releases it to activate voice commands
- The voice command application prompts the user via an audio cue to begin speaking
- The user says, “Call Jim on his mobile”
- The voice command system automatically detects when the user has finished speaking and begins the dialing process
- If any disambiguation is required (for instance, if there are multiple entries for “Jim”), the voice command system resumes listening without requiring another button push from the user

To detect when the user starts and stops speaking, the speech recognition technology had to perform a technique called “endpointing.” Endpointing had to be carefully implemented, in order to avoid interrupting the user when they pause briefly while speaking. Careful testing of the audio interface on the device was required. Not all speech recognition systems supported endpointing because of the complexity of the algorithms and the need for close integration to the device.

It was also important for these speech dialers to provide audio cues to the user for when they were not looking at the device. Audio prompts and high quality text-to-speech have been incorporated into some applications to provide audio confirmation of the name/number being dialed, and to disambiguate if there are multiple matches. For example:

User: “Call Jim on his mobile phone”
System: “Multiple matches found...Jim Ardman...Jim Smith...Jim Workman”
User: “Jim Smith”
System: “Calling Jim Smith’s Mobile Phone”

Text-to-speech must be used in this example to playback names from the contact list. If high quality text-to-speech is embedded on the device, then it can be used to enhance the minimal attention interface by performing services such as:

- announcement of incoming caller ID number or name
- announcement and reading of incoming text messages
- announcement and reading of incoming emails
- reading menus aloud

For the last several years, device manufacturers have been deploying the applications with phonetic speech recognition and high quality text-to-speech. One

example is the Nuance’s Vsuite product, which can support dozens of languages and contact lists with thousands of names. These applications perform best when integrated as a fully integrated capability of the device, in order to provide the best possible user experience.

2.2.3 Drivers Begin Talking to their Cars

Several years ago, auto manufacturers began putting computing platforms into cars to add new features to the car, including voice commands. Typical voice commands have included Bluetooth-enabled voice dialing, and voice control of in-car functions, such as turning the radio on/off, changing stations, changing CDs, or modifying the heat/air conditioning temperature settings. Text-to-speech technology has also been used to provide turn-by-turn driving directions for in-car navigation systems, as well as after-market navigation systems that can be installed by the car owner – like those offered by TomTom and Garmin. In recent years, navigation applications have even incorporated more sophisticated speech capabilities that allow users to enter destinations (addresses and points of interest) just by using their voice, with full step-by-step confirmation with the use of text-to-speech technology.

The automotive environment presents one of the most challenging environments for speech recognition. It is essential to minimize the visual and manual engagement required by the driver: there can be many passengers speaking in the car while commands are given, or there can be music playing, or there can even be simpler elements of background noise coming from outside, such as wind and other factors.

For these reasons, automotive manufacturers have invested in the optimization of speech applications for a specific car environment. They have incorporated high-quality built-in microphones and noise reduction technology. Applications were trained on audio data using the specific acoustic environment of the car.

2.2.4 Assistive Applications on Mobile Devices

Speech technologies have been used on mobile devices to enable and enhance service for blind and visually impaired users, as well as those in the disabled community. Common applications included:

- voice dialing with audio confirmation
- screen reading
- caller ID announcements
- reading incoming text messages and email

Assistive applications needed to consider the needs of the community of users carefully. For example, Nuance Communications TALKS screen reader for mobile devices included features for adjusting the volume and speaking rate of text-to-speech, and also included integration with external Braille input/output devices.

2.3 Speech Technology and the Data Network

As described in the previous section, speech recognition and speech synthesis can be performed on mobile devices with great success, and the technology has continued to get better from year to year. However, speech technology is hungry for CPU and memory cycles. The emergence of higher powered devices provides more processing power for on-device speech; however, these devices also come equipped with many new services such as web browsing, navigation and maps, and media players that consume resources – but do create a need for much more advanced speech recognition than traditional voice dialing or commands.

Fortunately, the availability and reliability of wireless data networks is rapidly increasing, and many of these higher-end devices are equipped with unlimited data plans. This creates a great opportunity for speech, allowing speech-based applications to take advantage of the data network to perform advanced speech processing on network-based servers rather than on the device itself. With network-based speech recognition, the audio is collected on the device by the application, and transmitted across the data network to specialized servers that perform transcription of audio to text and then sends the text back to the device. With network-based text-to-speech, the text is sent to servers and converted to audio which is streamed back to the device.

Network-based speech technology has several key advantages, namely,

- speech technology can take advantage of unlimited processing power in the cloud
- with this computing power, tasks such as speech-to-text transcription can be done very accurately
- some tasks, such as web search and navigation, can take advantage of data on the network to improve accuracy (web listings, address listings, movie names, etc.)
- the technology can be easily refreshed on the server side so that it stays up to date, whereas “factory installed” technology is usually not updated in the field
- speech that is processed in the network can help researchers improve accuracy of the core technology

There are, however, some limitations:

- Highly-used networks can introduce latency. If the network is fast and not congested, then results may typically be returned in a few seconds. However, if the network is slow or experiencing a high volume of usage, results may take much longer
- the data network is not yet highly available in all areas
- if the speech application itself is not factory installed, it may be more difficult to capture audio effectively and to integrate seamlessly with applications on the device
- some applications, such as voice dialing or music playing, if implemented on the network, would require that users upload personal data to the network

In the next 5 years, we can expect to see hybrid solutions that leverage both on-device and off-device speech resources performing most effectively in the mobile environment. For example, a device may leverage local resources for voice dialing and local music playing, and go to the network only when it needed resources for speech-to-text dictation or voice search.

2.4 Emerging Mobile Voice Applications

In the last few years, a variety of new applications for mobile devices have emerged that leverage network-based speech technology. In some cases, these applications have been made available for download to high-end smart phones such as iPhone, Blackberry, Android, Symbian, or Windows Mobile devices. In other cases, they are preloaded on mobile devices or into automotive platforms.

2.4.1 *Voice Navigation and Mapping*

Application providers that make navigation and mapping technologies have been among the first to incorporate advanced speech technologies into their applications. Speech technology is used to make input/output easier when on the go, or when using a small footprint keyboard or touchscreen keypad.

These applications can be enhanced by:

- entry of destination address by voice
- entry of landmark or point of interest by voice
- lookup business names or other content criteria (e.g., “Dave Matthews concert”)
- playback of specific turn-by-turn directions using text-to-speech

Implementing speech enabled navigation can be a complex task, especially for multilingual systems. Generic speech recognition technology alone is not enough. The technology must be trained on the “long tail” of addresses and locations for which people will need directions. Also, it is essential that the application support natural language interfaces, as users will have low tolerance for following several steps to input city, state, and to speak the names of businesses or destinations in a highly constrained fashion.

2.4.2 *Message and Document Dictation*

The emergence of text-messaging and email as popular mobile applications has been rapid, driven in part by the availability of full QWERTY keyboards on mobile devices. However, the keyboards are small and difficult to use for many users, touchscreens are difficult to use for typing, and it is impossible and unsafe in on-the-go situations.

For years, the dictation of text has been a successful application in the desktop and laptop world, with software like Dragon Naturally Speaking that is trusted and used by millions. Network-based computing power now makes it possible to perform speech-to-text dictation from mobile devices. Nuance has recently released a version of Dragon Dictation for the iPhone that provides a simple user interface for dictating text for email, text messages, social networking applications, and any application that requires text entry.

Dictation technology will work best when integrated into the applications that use dictation, such as email and messaging clients. On some mobile operating systems, such as Symbian and Android, it is possible to include speech as a universal input method that is active in any application that allows text entry. On feature phones and other operating systems, it may only be possible to include speech dictation by modifying the applications that need to use dictation to interact directly with the recognizer.

There are several important ingredients for success of speech dictation in mobile applications:

- the speech-to-text technology must be mature and robust for the language which is being dictated...it can take years of real-world use from a variety of human voices to make this technology robust
- the user interface must be clear about when and how to activate speech recognition
- ideally, the speech recognition technology can learn from the user's voice, common names they use, common terms used in their email and messages...this can require the user to give permission to upload some of this data to the network
- the user must have some way to correct mistakes; ideally, this will be a "smart" correction interface that gives the user alternate word/phrase choices so they do not need to retype

2.4.3 Voice Search

Similar to voice dictation, voice search allows the user to perform search queries using their voice. These queries could be:

- general search queries fed into a search engine such as Google, Bing, or Yahoo
- domain specific queries, such as searches for music or video content, product catalogs, medical conditions and drugs, etc.

For voice search to work well, the speech technology must be trained on common terminology that is used in search queries. A general voice search engine should know about celebrity names, top news topics, politicians, etc. A medical voice search engine should be trained on medical terminology and drug names.

Voice search has been built into many popular search engines. However, it may become more interesting as applications emerge that can determine the type of search and the user intent, and launch searches into appropriate content sources.

2.4.4 Speech Applications for Consumer Devices

Speech technologies have been deployed on a variety of mobile devices other than mobile phones and automobiles. Examples include:

- voice commands for portable gaming devices such as the Nintendo DS
- text-to-speech for reading content on mobile content readers such as Amazon's Kindle

- voice recognition and text-to-speech on MP3 music players to play artists, song titles, and playlists

2.5 Speech and the Future of Mobile Applications

2.5.1 *Enterprise Applications and Customer Service*

Enterprises, such as banks, mobile operators, and retail companies, have begun to invest in mobile applications. The rapid adoption of smart phones, such as iPhone, Blackberry, and Android-based phones, has provided a set of platforms for the development of downloadable applications that can reach a broad segment of the customer base.

Speech recognition provides many benefits to customer service applications today in over-the-phone voice response applications. These benefits can be extended to mobile customer service applications as well so that callers can speak to mobile applications in order to get product information, account information, or technical support. Speech can remove usability constraints from the mobile interface and allow enterprises to build more complex applications that provide better self-service capabilities.

Potential examples of speech usage would be:

- Using an open-ended “How can I help you?” text box at the beginning of the application that would enable the user to type or speak their question and then launch an appropriate mobile applet (a small application that performs limited tasks) that would provide service...instead of forcing the user to navigate through visual menus.
- Adding a product search box to a mobile application, and so the user could say the name or description of the product for which they need service.
- Speaking locations for store/branch locators.
- Speaking lengthy account numbers or product codes for service activation
- Dictating text into forms for applications (e.g., a mobile loan refinancing application).

Companies may find valuable use for mobile workforce applications, such as:

- Dictating notes into CRM applications
- Dictating notes into work order management
- Dictating into mobile order processing applications

2.5.2 *Natural Voice Control*

Now that it is possible to accurately convert speech-to-text using the computing power available via the data network, it is possible to take the next steps in voice control of devices and applications. Today’s voice command systems present a limited set of choices, and users must have some idea of the syntax used for commands.

As the number of applications and services available on mobile devices expands, it will be necessary to provide a more natural spoken interface for users, and to provide an intelligent interpretation of the user's intent. For example:

| User's request | Appropriate action |
|---|--|
| “Send text to John Smith...I'll meet you at Redbone's at 6pm” | Launch the text messaging client, address the message to John Smith from the contact list, and feed the text into the message client |
| “Find the nearest Mexican restaurant” | GPS locate the phone, launch the default maps/navigation software, and search for Mexican Restaurants |
| “Call John on his cellphone” | Determine if John is the only “John” in the contact list...if so, then place the call...otherwise prompt for more info |
| “Turn my Bluetooth on” | Activate Bluetooth |
| “How tall is the Eiffel tower?” | Launch a search application and feed it the text |

Translating the spoken words to text is the easy part; determining the actual intent for diverse populations of users, in a variety of languages, is the challenging part. Supporting this type of voice control system for a wide variety of global languages and cultures is also difficult. And finally, integrating voice control into a variety of applications on a wide variety of devices and platforms could be very difficult. However, the technical capabilities exist today, and so certainly mobile devices will evolve in this direction.

2.5.3 *The Future of Multimodality*

Predictive text, speech recognition, and text-to-speech software are already prevalent on many devices in the market. Other technologies are also emerging to make it easier to input or read text on a variety of devices, including:

- Continuous touch technology, such as Shapewriter, which allow users to slide their finger continuously around a touchscreen keyboard to type.
- Handwriting recognition technology, such as Nuance's T9Write, which recognize characters, entered on a touchscreen.
- Font rendering technology, such as Nuance's T9Output, which provide capabilities for more dynamic and flexible presentation of text fonts on mobile devices.
- Haptic feedback technology which provides vibration or other cues to the user.

Today, users typically must choose a particular mode of input or output. Traditionally, different input/output technologies have not always interacted seamlessly, though that phenomenon is starting to change, as some devices have begun to combine speech and text input in interesting ways. For instance, Samsung devices like the Instinct and the Memoir allow users to pull up the text input screen

with their voice, and automatically bring them into a touchscreen QWERTY text input field that features predictive text....however, users still find themselves either in speaking mode or typing mode, but not both at the same time.

There are situations where voice input is not appropriate or not feasible: in a meeting, or at a loud concert, for example. Similarly, there are situations where text input is not feasible or safe: driving a car, walking the dog, carrying packages. It will become increasingly important for input/output technologies to interact seamlessly based on user choice and preference.

For example, consider the following potential multimodal interactions, which could be implemented with technologies available today.

2.5.4 Multimodal Navigation

- The user presses a button and speaks a query: “Find the nearest coffee shop.”
- The application GPS locates the phone, and then launches a map application which presents a map of nearby coffee shops.
- The user uses his finger to draw a circle around the desired coffee shop...the mapping application zooms in on the desired area.
- The user presses the speech button and says, “Send this map to Mike Smith.”
- The email application launches, with a link to the map attached. At this moment, several people walk into the room. The user wants to communicate a private message, and so he uses predictive text technology on the touchscreen to type a message: “I will meet you at this coffee shop at 4:30 to finalize the sales presentation for Acme Corporation. I think we should lower our bid to \$400,000...give it some thought.” He then hits the send key.

2.5.5 Multimodal Service Calls

- The user gets a text message from his airline that indicates his flight has been canceled, with a link to a number to call to rebook his flight.
- The user clicks the link and places a phone call and is connected to a service agent, validating the call is from his mobile device.
- The service agent uses a data application to push a set of alternate flight options down to the user’s phone. An application framework on the phone launches while the user is still on the call with the service agent.
- The user can use the touchscreen to scroll through options and look at layover times, seat availability, and arrival times.
- When the user determines the desired flight, he selects the flight.
- The service agent completes the change, and then pushes a boarding pass to the user’s mobile device which can be scanned by the user at the airport.

2.6 Looking Forward

There is no doubt that speech technologies will continuously evolve and provide a richer user experience, enabling consumers to leverage the input and output methods that are best suited for them moment to moment. The key to success of these technologies will be thoughtful integration of these core technologies into mobile device platforms and operating systems, to enable creative and consistent use of these technologies within mobile applications. Continued emphasis on the user experience will also be key, to ensure that users understand where and how to speak to mobile devices in a manner that is successful.

Chapter 3

“Why Tap When You Can Talk?”: Designing Multimodal Interfaces for Mobile Devices that Are Effective, Adaptive and Satisfying to the User

Mike Phillips, John Nguyen, and Ali Mischke

Abstract It is becoming clear that as mobile devices become more capable, the user interface is the last remaining barrier to the scope of applications and services that can be made available to the users of these devices. It is equally clear that speech has an important role to play in removing these user interface barriers. Vlingo, based in Boston, is a four-year-old company that creates multi-modal interfaces for mobile phones, by making use of advanced speech technologies. Our chapter discusses the opportunities and challenges that are presented in the mobile environment, describing the approaches taken by Vlingo to solve such challenges. We present findings from over 600 usability tests in addition to results from large-scale commercial deployments.

Keywords Multimodal user interface for mobile phones • Mobile speech interface • Natural Language dialog • Information retrieval • Out-of-grammar failures • Mobile use while driving • Mobile search • Mobile messaging

3.1 Introduction

Across the world, network-connected mobile devices, including phones, laptops and PDAs, are quickly becoming our primary sources of communication, information and entertainment. This transition has been driven by both the attractiveness to end consumers and by the continued advancement in processors, memory, displays and wireless data network capabilities. Modern mobile phones (even low cost versions) now come with more processing and memory than desktop PCs from a few years ago. They also have bright high resolution color displays, and can connect to services through high bandwidth data networks. At the higher end of the market, so-called

M. Phillips (✉)

Chief Technology Officer, Vlingo, 17 Dunster Street, Cambridge, MA 02138-5008, USA
e-mail: phillips@vlingo.com

“smart phones” now come with fully capable operating systems that allow users to install various applications from an exciting marketplace.

Because of these advancements, people are relying on mobile phones more and more in their daily lives. In fact, in parts of the world where PC and internet penetration is still low, mobile phones are leapfrogging the adoption of PCs and laptops, frequently serving as users’ only access to network connected information and applications.

Given such rapid advancements, the only remaining barrier to what can be done on mobile devices is the *user interface*. Because of the inevitable size constraints, mobile user interfaces have been limited to relatively small displays for output, and small keyboards and touch screens for input. Although applications that require only simple interactions can fit easily within these confines, there are many applications (especially those which require significant amounts of text input) that are difficult to use given such constrained interfaces. Moreover, this problem of constrained interfaces intensifies with the use of mobile devices when driving a car, a situation which itself presents a whole other set of issues, mainly related to safety.

Clearly, speech has an important role to play in creating better interfaces on small mobile devices. Not only is speech the most natural form of communication for people, it is the only interface for mobile devices which is not constrained by form factor. Even so, it is also clear that speech interfaces should be developed in conjunction with other modalities. People can speak much faster than they can type (especially on a small mobile device), but can read much faster than they can listen. One must also take notice of the fact that while there are many situations where it would be safer and more convenient to speak and listen, there are indeed those situations where it would be inappropriate to use a speech interface altogether. So, the overall mobile user interfaces needs to support speech along with other modalities of input and output, and allow the user to freely and easily switch between modalities depending on their preference and situation.

While the benefits are apparent, creating usable speech interfaces presents significant challenges both from technological and human factors perspectives. This is why, notwithstanding significant investment in this technology over decades, speech interfaces have remained constrained in both their functionality and market success.

3.2 Users and Goals

Any examination of users’ experience with a technology must begin with an understanding of the key users and their relevant goals. At Vlingo, we find most mobile speech recognition users represent one of four personas, or user segments:

1. Pragmatic Users
2. Social Users

3. Stylists; and
4. Technophiles

Overlaid onto those four user types is the set of blind, low-vision, or physically disabled users for whom accessibility is of primary concern. We will touch briefly on accessibility usage here, but leave a more extensive discussion and analysis for a separate discourse.

3.2.1 Pragmatic Users

While pragmatic users may own any mobile phone, this persona is most commonly associated with the usage of a BlackBerry® or Windows Mobile device in the US market, and Symbian E-series devices in other markets. The pragmatic user, often a “prosumer” (professional/consumer hybrid), is primarily interested in using a mobile phone to transact professional and personal business on the go, employing email, navigation and web search, among other mobile applications. When pragmatic users evaluate any potential solution, efficiency and convenience are the most important considerations. This user is comfortable using technology as a tool, but is not interested in technology for its own sake.

3.2.2 Social Users

Social users choose a device and a set of applications to help them keep in touch with friends and manage their active social lives. The stereotypical social user owns an iPhone, although this group is represented among owners of perhaps all mobile devices. The social user is often active on Facebook and/or Twitter and sends and receives an often staggering number of text messages. Social users get excited about any solution that makes it easier and more fun to make plans and share day-to-day information and media with friends.

3.2.3 Stylists

Stylists are fashion-conscious users who see their phones not as a tool, but as an extension of their image. These users invest the most time and money in personalizing ringtones, wallpaper, physical cases and other aspects of their mobile devices. In the US, many stylists are younger users (18 and under); in other countries, such as Italy, this segment may be much more highly represented among the general population. Stylists are not particularly interested in productivity; instead, they will try a new technology if it makes them look “cool” or helps them be more publicly visible.

3.2.4 *Technophiles*

The technophile represents the earliest segment of the adoption curve: the first user of any new system. Any first version of a market-changing technology must entice and delight the technophile, who evaluates many offerings and keeps only those that prove to be particularly useful or fun. Technophiles are the vocal fans that support and nurture a new product, and partly for that reason they are likely to serve as the first usability and beta testers that evaluate and help refine a technical offering.

Many technology companies find it easiest to design for technophiles, as this persona is often overrepresented within the company's own walls. Technophiles are critical to a company's entry into and continued success in the marketplace; however, a mobile consumer offering that caters only to this persona at the expense of mainstream utility will ultimately stall. Indeed, while "cool new technology" drives the technophile's decision on what to evaluate, even the technophile will quickly abandon a product if the technology is not also useful, practical and/or fun.

In truth, virtually no user outside the walls of a speech recognition company would approach an application purely with the goal of "speaking to send a text message" or "speaking to search the web," and certainly no user would articulate as a goal the desire to speak to fill in a text box. Instead, a typical user seeks to "text my friend and tell her I'll be 5 minutes late" or "find the address of the restaurant without taking my eyes off the road."

As we review the current state and the future of mobile speech recognition applications, it becomes important to revisit this concept of the user's *true* goals when compared with the development team's list of product requirements.

3.2.5 *Accessibility Considerations*

The challenge of interacting with applications through a small keyboard and display can be especially difficult for people with various physical or sensory impairments. Speech-enabled interfaces, both for input and output, may be able to help such users interact with various applications and services on mobile devices. Although people with one or more impairments can also be categorized according to our personas described above, they are most likely to be more concerned about the degree to which the interface can help them use their mobile devices than anything else, and so are more biased towards the needs of the Pragmatic User.

For example, we have been contacted by a number of deaf users who use the speech-to-text functionality of our system to allow them to communicate with people who do not know how to produce sign language. So, they will hand their phone over to the person they are trying to communicate with who will then speak into an application such as a notes page so that the hearing impaired person can see the text. While we did not design our interfaces with this usage in mind, it is gratifying to see that we are able to help broaden communication possibilities for people with disabilities.

3.2.6 Mobile Use While Driving

Similar to disabled users, the other common usage scenario which cuts across the four personas listed above, and comes with some specific needs, is that of *mobile use while driving*. In a recent study of mobile messaging habits, we surveyed almost 5,000 people and asked them various questions about their use of mobile text messaging (SMS). One of the most dramatic findings is that over a quarter of all respondents admitted to texting while driving, although 83% of respondents think that it should be illegal. Of particular concern from a safety point-of-view, almost 60% of respondents aged 16–19 and 49% of respondents aged 20–29 admit to texting while driving.

While we see the beginning of legislation aimed at reducing this problem, it is well known that these are very difficult laws to enforce. In our study, we found no correlation between legislation and the percentage of people who admit to texting while driving when comparing rates of texting while driving in states with laws against this practice with those states that have no legislation prohibiting texting while driving. In addition, while there has been increasing attention to the dangers of texting while driving, we saw very little change in user behavior between studies we performed in 2008 and 2009.

We also see in our user surveys that almost half our usage of text messaging consists of people using mobile devices while driving their cars (see Fig. 3.1 below). Undoubtedly, users are attempting to reduce the risk of using their mobile devices in their cars by using speech interfaces. Given this fact, here at Vlingo we

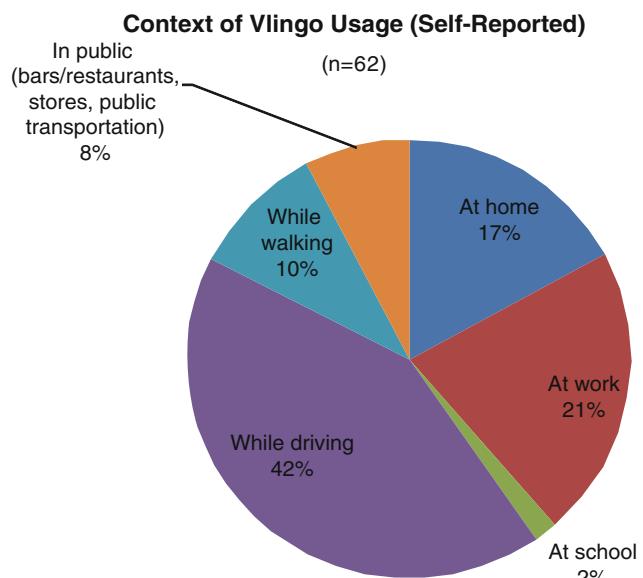


Fig. 3.1 Self-reported context of use

are increasing our focus on the user-interface needs of this use case. In particular, we are reducing the reliance on the screen and buttons on the handset by reading back input using text to speech (TTS), and allowing users to speak “send” rather than push the “send” button, among other things. While this added speech utility certainly increases the challenge of developing the user interface, we can focus our efforts on the particular tasks that we know users tend to perform while driving: phone dialing, messaging and entering destinations in a navigation system.

3.2.7 Additional Use Cases

While the driving use case is clearly important, we also find a significant percentage of self-reported usage in contexts such as home and work. In these contexts, hands-free and eyes-free interaction may be desirable, although it is not essential to complete the task. Interestingly, we find users attracted to voice-based interaction at home and work because of convenience and overall usability of the mobile device. That is, rather than having to find and navigate to a particular application and type in the desired content, it is significantly easier and more convenient to press one button and speak a single sentence. The relatively high usage at work and in public also point to a “cool” factor; some users, particularly technophiles and stylists, like to show off the power of Vlingo’s voice user interface as a measure of social status. The usage statistics also imply that while some users are understandably concerned about others’ overhearing their messages and searches, a non-trivial segment of the target user base is completely comfortable with using speech in a public setting for at least some tasks.

3.3 Existing Speech Interfaces

While there has been ongoing research on fully natural spoken dialog systems for many years, designers of most successfully deployed speech interfaces have taken the approach of tightly constraining what can be said and constructing the application around these constraints.

These highly constrained systems tend to employ a distinctive formula:

1. **Constrain the speech recognition task as much as possible.** By narrowly focusing what the application can accomplish, constrained systems can employ application-specific grammars, or specific commands or “key words” that the user must memorize, as well as application-specific statistical language models to predict user inputs.
2. **Construct the application around those constraints.** Through careful design of the application flow and details of the user interface design, try to elicit responses from users which match the constraints of the speech recognition system,

while still allowing them enough flexibility to perform the tasks that they are interested in.

3. **Require integration of semantic meaning to speech recognition.** Attempt to perform natural language processing or meaning extraction as part of the speech recognition task, such as mapping an address to street number, street name, city, and state.

This approach can be seen in various speech applications, including automated telephone-based customer services, devices built into cars and a few key applications on mobile phones, most notably voice dialing. If the system is carefully designed and tuned, users can experience reasonable rates of success.

But, despite the market success of these systems, we find that users tend to not like these systems. Such systems are perceived by users as being inflexible and error-prone, even in the cases where the system is eventually able to satisfy the task it was designed for. This perception is likely due to a number of well-known problems with the highly constrained approach that prevent it from being successfully applied to various mobile applications. In particular:

1. **User training.** Constrained speech interfaces work well only if the user knows what to say and is willing and able to present information in the expected order. Unfortunately, we know that most users have trouble staying within the constraints of the application and the speech recognition system. Users may forget what features can be activated by voice, may not remember what commands they can speak to activate those features or may simply want or need to provide information in a different order or perform some task, which is not supported by the constrained flow.
2. **Severity of out-of-grammar failures.** Even in successful telephone-based customer service applications, out-of-grammar utterances present a much greater source of errors than their in-grammar counterparts, and are much harder to recover from than speech recognition errors.
3. **Failure modes for out-of-grammar input can be very bad.** Since users do not know what is “in grammar,” they cannot distinguish the case where they are speaking something the system knows about but did not recognize from the case where they are speaking something the system simply is not equipped to handle. Since users do not understand the cause of these errors (or even worse, why they are suddenly in some unknown state of the application), it is difficult for them to learn how to modify what, or how, they are speaking to create a successful interaction. If its an in-grammar error, they may be able to recover by trying again and perhaps speaking more clearly. If its an out-of-grammar phrase, they could speak it all day long and the system would never get a correct response.
4. **Unnatural user interfaces.** Usage of grammars encourages applications to be developed with a sequential user interface when a one-step interface may be more natural. For example, directory assistance applications generally force users to say the city and state in one utterance, followed by the business name in the next utterance. This allows the application to switch over to a city-specific

grammar for business names, but conflicts with the user preference to say the entire request in one utterance.

5. **User patience and compliance.** Users are not trying to successfully navigate a voice activation system: they are using a voice-activated tool to complete a particular task. In a situation where the user has multiple choices of how to achieve that task, patience will be low if a system does not work as expected. Even in systems with carefully designed prompts, users do not always pay enough attention to the order in which information is requested, or may not have the required information available in the order or format the system requires.

Rather than invest a lot of time in learning how to use a voice-activated system, a user is likely to escape to an agent (in the case of phone-based systems), to typing (in the case of on-device applications) or to a competitive application (when available) if his/her first few attempts are not properly understood.

Although significant, these issues could be overcome if the goal was simply to allow the use of speech interfaces for a few key applications – users can eventually learn how to successfully interact with such applications if they are sufficiently motivated to use the speech interface. However, if the goal is to support the full range of applications that users may have on their phones, we cannot expect users to learn what they can speak into each and every state of each application.

In addition to these key usability issues, it is also the case that mobile application providers do not want to construct their applications around the constraints of a speech recognizer. There are now well over 100,000 applications available on one of the most popular mobile phone platforms – it would be too much of a burden for each of these applications to be designed based on the constraints and meaning-extraction semantics of a speech recognizer. Even if the intent were there, the vast majority of individual mobile application developers do not have the relevant domain expertise or available resources to handle the grammar development, speech user interface design and the ongoing recognition and grammar tuning activities required for such a system to succeed.

3.4 Natural Language Dialog Approaches

The field of speech technology has witnessed two challenging decades of work on combining speech recognition with natural language dialog processing to create automated systems, which can communicate in a more human-like manner.

Obviously, if we could really achieve the goal of creating automated systems which have human-level spoken dialog skills across a broad range of domains, this could be used to create highly functional user interfaces (since humans mostly succeed in communicating with each other). Unfortunately, if we only get partway to human-level performance, the interfaces become even harder for people to use than the more constrained interfaces. There are two key reasons for this: boundary-finding and efficiency.

3.4.1 Finding the Boundaries

For any user interface to succeed, the users need to have a mental model of what the system can and cannot do. For simple constrained systems, the system can make this obvious to the user by asking very specific questions (“what city?”) or by telling the user their choices (“say either send, delete, or forward”). On the other extreme, if we could make a system understand everything a person could understand, users could learn that they could talk to the machine in the same way they would speak to another person.

The problem is that if you make the system understand a lot of the things a human could understand, but not everything, how do you make this apparent to the user? How can the user know the boundaries of what the system can and cannot understand? This problem is not just limited to the words and sentences that the system can interpret, but extends to dialog constructs as well. That is, when people talk with other people, they do not just respond to individual utterances, but rather make use of a tremendous amount of shared knowledge about the current interaction, state of knowledge of the other party, and knowledge of the world. Unless you can fully simulate this in an automated dialog, how can you give the user a reasonable model of what the automated system can and cannot handle?

3.4.2 Search Efficiency

In the absence of the deep contextual information present in human-to-human communication, natural language dialog systems are necessarily inefficient. While this can create an annoyance for search tasks, it becomes completely untenable when applied to all but the simplest messaging tasks, particularly when we consider the fact that a user may need to correct some of what is recognized by the system. In fact, even when humans speak on the phone they do not necessarily recognize each other 100% of the time, so how can we expect machines to do so?

Consider two cases: finding and calling a restaurant; and composing an email. Although the dialog examples, provided below, could be optimized with careful design, they illustrate nonetheless some of the requisite complexity of systems that rely solely on natural language dialog.

For performing a search, the user may say something like “search vegetarian restaurants in Boston Massachusetts”. The first complexity arises when we consider that although a user thinks about *that great veggie restaurant in Boston* going from more specific to more general – the search database may require filtering in the opposite order: first narrowing down the state, then the city within the state, then the category of restaurants and then the name of the restaurant. Unfortunately, technology too often requires the user to conform to the system’s view of the world rather than alternatively adjusting the system to conform to users’ expectations.

Even if we consider a system that can handle the more specific information first, we are not yet out of the woods, as there happens to be more than one vegetarian

restaurant in Boston. In this case, the system may respond with, “Found four restaurants: Grasshopper, Grezzo, Peace ‘o Pie, Wheeler’s Cafe.”

User: “Grasshopper”
 System: “Great. What do you want to do?”
 User: “Call listing.”
 System: “Calling Grasshopper”

This dialog could perhaps skip a step: with careful design, the system could potentially have enabled users to say “Call <listing>” when the listings are first read back. If the desired listing was not in the initial result set, either due to misrecognition, omissions in the search catalog, or user error (“oh, wait...the restaurant is actually in Somerville!”), another step or two would be required. If the list of matching restaurants is too large to iterate through, then further dialog disambiguation would also be required. Furthermore, a linear dialog system limits the ability for users to choose among the restaurants by various metrics such as distance, rating or match to what was requested. Overall, the search case involves a tolerable number of steps, but does not compare either in efficiency or in user comfort to turning to your driving companion and saying, “Hey, can you call that veggie restaurant we like downtown?”

3.4.3 Messaging Efficiency

The messaging problem is vastly more complex, even if we make the simplifying, though not always satisfying, assumption that a user is sending a message to only one person at a time. In the messaging use case, users need to select a particular contact or specify new contact information, choose which contact information to use if there are multiple phone numbers or email addresses stored for the given contact and compose a completely unconstrained message. In the case of email, the user must also differentiate the content for the subject line from the content for the message body. Composing the message and verifying that the content is correct is a complex problem not only for the speech recognition system but also for the user.

Imagine the user said something like, “Email John Smith subject Saturday afternoon message let’s climb mount Osceola I’ll pick you up at four.”¹

In this case, a system that provides only dialog rather than multi-modal feedback places on the user a high cognitive load (defined as the burden placed on a user’s working memory during instruction). In the midst of whatever else the he or she may be doing (driving, making a mental “To Do” list), the user must listen carefully enough to answer the following questions:

¹Note that in these examples we are not including punctuation. This is because we are attempting to show the input to the system. While our system does has the ability to insert punctuation for things like email dictation, the spoken form from the user generally does not include any indication of punctuation, so this is what we show in these examples.

1. Did the system understand that I am trying to send an email?
2. Did the system choose the right contact?
3. If I have multiple email addresses for John Smith in my address book, did the system choose the desired address?
4. Did the system properly parse the content into subject and message?
5. Was the content recognized accurately?
6. Did the system correctly understand words that have homonyms (e.g., four vs. for)? More simply, were any words mistaken for other similar-sounding words that would be difficult for a listener to differentiate?
7. Was capitalization and punctuation added correctly?
8. What about local or technical terminology? Does the system recognize Mount Osceola in New Hampshire? If the system repeats a word that sounds wrong, can the user differentiate between a recognition error and a word that the text-to-speech (TTS) engine has not been tuned to pronounce properly?

Even in the case where everything is recognized correctly, and setting aside relatively rare events such as the use of homonyms, it may not be reasonable to expect users to pay sufficient attention to verify multi-field content aurally. In cases where users are driving or multi-tasking, they may easily miss part of the readback.

Additionally, users sometimes compose messages in multiple utterances, supplying some content, taking a breath to gather their thoughts before completing the message in a second utterance. A dialog-based system does not preclude this behavior: the final prompt could say something like, “Do you want to send or add to your message?” However, it is certainly more efficient to simply start speaking the remaining content rather than reply, “Add to my message,” then wait for a prompt indicating that recording is ready to begin.

Now, in the scenario above, imagine that some part of the user’s speech was mis-recognized. The error could be material (for example, the system chose the wrong contact or the original meaning is no longer from the recognized text), or could be immaterial to the message semantics (such as a singular/plural error or an added or dropped article). Of course, there could also be multiple errors of varying severity.

Although users do not expect perfection in their text messages (and they rarely *type* error-free text messages!), they do appear to consider email a more formal medium and therefore expect a noticeably higher degree of accuracy. And, of course, in any messaging medium, the message must be addressed to the right contact and the meaning must be understandable to the recipient.

The complexity of correcting speech recognition errors in a spoken email rapidly mounts when you consider the user’s need to identify which piece is wrong, supply the new content, and verify the full message before sending. It is not difficult to imagine a user’s growing frustration when trying to navigate this complexity using a dialog-based system. The user could certainly simplify correction by starting the entire task again, but this will be accompanied by a loss in confidence that can soon lead to the user abandoning the system. At some point, likely sooner than speech technologists might prefer, the user thinks of the system as unreliable and starts to believe it is easier to type.

3.5 Text-Based Information Retrieval

In many ways, text-based information retrieval shares many of the same issues as spoken-language interfaces. To allow users to find information from a large set of possible sources, how can you allow users to express some complex set of constraints to navigate through large numbers of possible results?

There can be either very structured ways to do this (analogous to command-driven speech interfaces), or more “natural language” driven approaches. So, systems that can accept natural language queries perform deep analysis of these queries, and engage in a series of further dialog steps to narrow down the choices and finally present sets of results for the user.

Of course, the approach that has gained widespread market acceptance is keyword driven search, driven not by natural language analysis, but rather by algorithmic search that relies on the underlying data to guide the search to the most popular results.

You can certainly argue that this approach is overly simplistic and that it would be possible to produce better results by making more use of language and dialog – asking users, for example, for clarification to help narrow down results.

However, based on the market success of search engines, it seems that these potential benefits are outweighed by the simplicity of web search. Users quickly learn a very simple model of interaction – they type in a few words, get a list of results, and if they do not like what they see, try some different words. They do not need to learn some more complex model of interaction, and they do not need to worry about the boundaries of what words and language constructs the system can understand.

3.6 Unconstrained Mobile Speech Interfaces

How can we apply this same notion to mobile speech interfaces, and avoid the spoken dialog problems discussed above?

The key characteristics of open web search which has allowed it to succeed over previous approaches are:

1. A very simple interface (type in some words, see some results)
2. No boundaries to what you can type. Thus, users do not need to worry about what they can type and they do not need to learn new interfaces for each type of thing they are searching for.

In our work at Vlingo, we have been making use of these principles in designing a broad speech-driven interface for mobile devices. Rather than build either constrained speech-specific applications, or attempt to make use of more complex natural language dialog approaches, we have been working to create a simple but broad interface which can be used across any application on a mobile device.

3.6.1 User Experience Guidelines

Our efforts to create a simple, transparent model for the user have resulted in the following product principles:

- 1. Provide multi-modal feedback throughout the recognition process:** Using a combination of graphical and audio feedback, we let users know what action will be taken. Users are most interested in voice when their hands or eyes are otherwise occupied. As a result, a good speech recognition system provides a combination of tactile, auditory and visual cues to keep the user informed of what is happening. When the Vlingo user first presses the voice key, we display a Listening popup, and reinforce the display with a vibration or ascending tone (depending on platform). Accordingly, when we finish recording and begin processing audio, we change both the wording and color of the on-screen display and play either another vibration or a descending tone. This feedback is useful on all platforms, but is especially important on touchscreen devices, where users do not have the immediate haptic (also known as “touch-based”) feedback of feeling a physical key depress and release.

When the user’s recognition results are available, we play a success tone and provide text-to-speech confirmation of the task that is being completed. Text-to-speech allows the user to confirm without glancing at the screen that we understood their intention. In this way, a user who may be multi-tasking is alerted to return their attention to the task at hand. Finally, in cases such as auto-dialing, where we are about to initiate a significant action, we show a temporary confirmation dialog, play a variation of the success tone and again use text-to-speech to confirm the action we are about to take. When we have correctly understood the intended contact, the pop-up, tone and text-to-speech provide assurance; in the case of misrecognition, the multi-modal feedback calls the user’s attention to the problem so they can correct their entry before we initiate the action.

- 2. Show the user what was heard as well as what was understood:** When a user speaks, we show the user what was recognized, how the system interpreted that speech and what action will be performed as a result. Traditional IVR-descended mobile applications show how the system interpreted speech but do not show exactly what the system heard. This can cause confusion in cases where misrecognition results in an unexpected and undesirable action to be taken.

Consider a user requesting “sushi” in a local search application. The system might have heard “shoes” and would return the names of local shoe stores. A user seeing the name of local shoe stores when searching for sushi is unlikely to make the phonetic connection, and may not understand why the system is displaying local shoe stores. The user would not be able to distinguish whether the system misrecognized what was said, or whether the search results were wrong because there were no sushi restaurants available.

In contrast, Vlingo shows recognized text in a standard text field along with the system’s interpretation of that text. In the example above, the user would see the word

“shoes” in the search text field and would be better able to understand why the top listings included shoe stores like Aldo and Payless (which may not even have the word “shoes” in the business name) rather than the expected list of local sushi restaurants. The presence of recognized text helps to clarify how the system chose its match, making any misrecognition appear less random.

Additionally, there are cases where the user’s speech is correctly recognized by the system, but for various reasons, the search engine provides unexpected results. For example, a user may not remember where a particular business is located, and may search for it in the wrong town, or a user may simply use search terms that the search engine interprets differently than the user intended. Here, again, by showing the exact words that Vlingo heard, we help the user realize how to proceed. If the words were correctly recognized but the search engine does not return the desired results, it is clear that the problem is not one of speech recognition. Speaking the same words again will not help; rather, the user needs to modify the terms of the search.

The two examples above describe different types of errors: recognition errors and search errors. Showing exactly what Vlingo heard helps a user to understand what type of error has occurred, which is critical to the user’s ability to fix the problem quickly and complete the task at hand.

3. **Allow the user to edit results:** When faced with speech recognition results, the user can perceive the results as correct, incorrect, or almost correct. Instead of repeating the task for the almost-correct case, the user may choose instead to invest in their previous effort by correcting the results. Depending on the task and the user’s level of expertise, the user may choose to correct recognition errors by speaking again, editing by typing, selecting from alternate speech-recognition results, or some combination of these methods.

These mechanisms are mainly used to correct speech recognition errors, but they also handle cases where users make mistakes or change what they want to do.

4. **Preserve other input modalities:** As a correlate to the principle above speech is one way for the user to provide input, although users should be able to use other input mechanisms in cases where speech recognition is not practical or where speech recognition is not working well. Therefore, speech recognition should augment rather than replace the keypad and touchscreen. If speech recognition does not work well because the user is in a noisy environment, or if the task is easier to complete by pushing a button, the user has the option of using other input modalities.

Traditional IVR-descended applications require users to speak again in the case of misrecognition, as opposed to Vlingo’s model of displaying results in an editable text field. In the traditional model, users can lose confidence: Why trust that the system will correctly understand a re-utterance if the first attempt was not successful? If a second attempt is also unsuccessful, the user may abandon the task, or even the application, deciding it is easier to type.

Our model of providing recognition results in a fully editable text field that allows users to correct errors in the mode they prefer: speaking again; choosing from a list of options; or using the familiar keypad interaction. This correction ability,

particularly when paired with an adaptive loop that enables Vlingo to learn from successes and errors, increases user confidence in a voice-based system.

5. **Allow the user to add to results:** When composing a message, users often need time to gather their thoughts. It is relatively common for users attempting to speak a text message to speak the beginning of their message, pause for a few seconds while they decide what else to say, and then complete their dictation. It is also not uncommon for a user to reread what they have dictated and decide they want to say more. The multi-modal nature of our approach makes this use case easy to support. For messaging tasks, we place the cursor at the end of the recognized text. Once users see what we recognized, they can instantly initiate a new recognition and append text to their message.
6. **Give users explicit control over action taken:** The action which is taken depends on the state of the mobile phone. In the case where there is currently an application in the foreground, the action taken is very simple: we fill the current text field with whatever the user just spoke. Hence, in this case, the model of the speech system is very clear to the user; it is just an alternate input method, so it acts like the keyboard. Any constraints in what makes sense to speak are imposed from the application – just as if the users were typing into the current text field.

However, we believe that speech interfaces can serve a function beyond simply replacing the keyboard; they can also be used to help users navigate through the various applications available on their phone. Thus, in addition to acting like a keyboard to allow users to fill text fields, we also handle high-level application routing.

Here is a case in point: If there is no application currently in the foreground, and the user says something like “send message to Joe thanks for sending me the new application” the desired action is clearly to start a messaging application, fill in the “to” field with “Joe” and fill in the “message” field with “Thanks for sending me the new application.” To maintain the principle of letting users know what was said and giving them ways to correct it, we first bring them to a form which includes the action which will be performed along with the contents of the text fields to be used.

A similar use case involves the use of commands inside an application, such as the user saying “forward to Joe” when reading an email. Again, to let the user know what was said and potentially correct it, we bring them to a form that shows the recognized command with text-field contents. In the particular example of “forward to Joe”, the form can be the same screen as if the user had selected “forward” from a menu and typed in “Joe”. From that point, the user can confirm by saying “Forward” or click on the “Forward” button. By doing this, we’ve given the user multiple modalities to start and complete the task.

While this is indeed a more complex user model than simply acting like a keyboard, we find that the benefits of this top-level application routing are worth the added complexity.

7. **Ensure the user is aware of the system’s adaptive nature:** The most common user complaint about any speech recognition system is that it is not accurate enough.

Indeed, the industry may never rid itself of that complaint: as technology advances, so do user expectations.

However, Vlingo includes an adaptive loop based on acoustic and language characteristics of the user and of all speakers of the user's language. This component continuously improves the models of the system based on the ongoing usage – including any corrections that users make to the system's responses. The inclusion of an adaptive loop is critical to users' satisfaction; equally important is the user knowing about that adaptation up front.

While Vlingo performs quite well from the initial utterance, the user's adaptation to the mobile device begins to improve performance after 3–5 utterances. Not surprisingly, we have found that those users who are dissatisfied with their recognition results are significantly more patient and more likely to continue to use Vlingo if they are told that speech recognition improves over time than if they believe their initial results are the best to be expected.

There are several possible factors in operation here; we believe all have some contribution. Most obviously, users who are told the system will improve over time tend to believe what they are told. If their initial experience was not satisfactory, they will try again in hopes of realizing the goals of improved usability, speed and convenience that originally led them to try the software. Additionally, early adopters are interested in the technology and intrigued by the concept of the system's adaptability. These users, if not satisfied with initial recognition results, are more likely to play with the software if they believe it will adapt simply so they can try to understand how the adaptation works.

As users begin to use any system, their success rates increase as they gain increasing mastery over the system's interfaces while at the same time learning of its limitations. Users usually do not invest the time to analyze whether their increased success is due to their own behavior changes, or due to the system actually getting better at understanding them. However, by informing users up front of the adaptive nature of the system, most users tend to give the technology credit for at least a portion of the positive results.

Finally, users appear to enjoy engaging with a system they believe to be intelligent. If a system promises to learn, users are motivated to help the system do its work: correcting errors, adjusting how they speak, and generally giving the system more time to learn the user's voice. Essentially, a spirit of cooperation develops between the user and the system's adaptive loop.

3.6.2 Commercial Deployments: Guidelines in Practice

We have now deployed in the past two years a number of commercial systems based on these principles, including:

- We worked with Yahoo to deploy a speech-enabled version of OneSearch, Yahoo's mobile search product. This was first deployed on Blackberry devices

and allows an unconstrained and multi-modal input within the Yahoo search application;

- We released our first top-level voice user interface, first on BlackBerry devices and then on iPhone. This application allowed users to speak top level commands (“send message to Joe let’s meet at Peet’s,” “find restaurants in Cambridge”, etc.) plus provides a full multi-modal interface within each of a known set of applications (phone dialing, SMS, Email, social network updates, web search, notes);
- We expanded this to a broader set of devices including Nokia phones running the Symbian Series-60 operating system as well as phones running Microsoft’s Windows Mobile operating system; and
- We added the functionality (called “Vlingo Everywhere”) which allows users to speak into any text field on BlackBerry devices. So, we have been able to hook into the operating system such that if the user speaks when a text entry field is in focus, we will paste the text into the text field – without any modification to the application which is receiving the text.

These commercial deployments have been constrained by the functionality exposed by the existing mobile operating systems, and thus have not yet allowed us to fully implement the broad interface described above. Nonetheless, as mobile operating systems become more open, and start to be designed with speech input in mind, we expect to soon be able to provide such a broad interface.

3.6.3 A Tour of Vlingo for BlackBerry

With Vlingo for BlackBerry, users can speak to dial a contact, search the web, send SMS and email messages, update Facebook and Twitter status, create notes to self, open applications, and speak into any text field on their device. To initiate voice, users simply press a side convenience key. The presence of a physical key allows the user to activate Vlingo at any time, eliminating the extra steps inherent in navigating to and opening an application.

If an application is in the foreground when the user begins to speak, Vlingo will paste text into that screen. In Fig. 3.2, for example, the user is speaking to add a calendar entry using the BlackBerry’s native calendar application. By default, we show a pop-up for the *Vlingo Everywhere* feature, allowing the user to add or correct text before passing it off to the destination application. Today, the BlackBerry platform precludes us from including word-level correction within native applications. In the future, if speech becomes more tightly integrated into the platform, we can eliminate this extra step. Until then, we do allow users to turn off the pop-up and inject text directly into 3rd party applications if desired.

To take advantage of Vlingo’s routing capabilities and start a new task, the user can either begin on the phone’s idle screen or say “Vlingo” before their command from any screen. For example, in Fig. 3.3, “Vlingo, search yoga classes in Cambridge Massachusetts.” Vlingo allows the user to choose a default search

Fig. 3.2 Speaking into the native calendar application

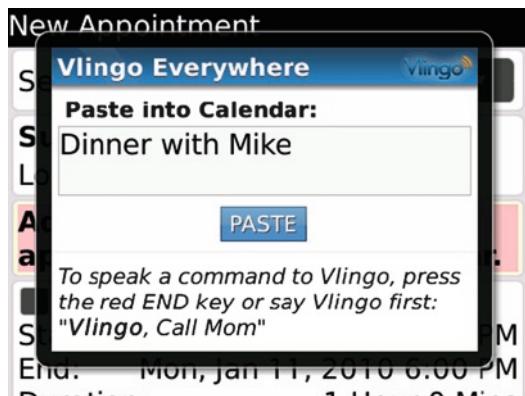
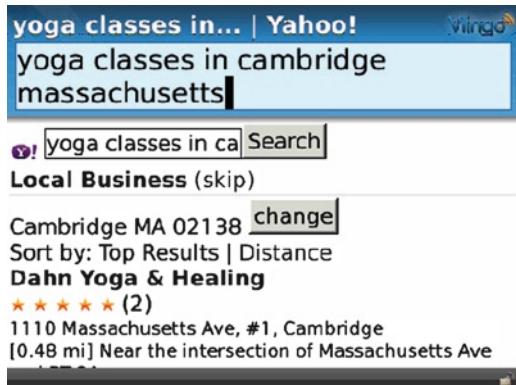


Fig. 3.3 Web search from the top level



engine (Yahoo or Google), and displays recognition results and search results on the same screen. In this way, the user can easily tweak and re-execute the search, either if a word was misrecognized or if the search results were not satisfactory.

As shown in Fig. 3.4, Vlingo saves even more work for the user during complex tasks such as SMS and email. In a single utterance, the user can say, “**Send message to John message let’s meet for pizza at 7.**” “Send message to” is just one of myriad ways to alert Vlingo that you want to send a text message; the application is flexible enough to understand various permutations a user might reasonably speak, and has the ability to adapt to countless other permutations as the vernacular evolves.

In the text messaging case, Vlingo recognizes that the user wants to send a text message, maps “John” to a particular name in the address book, recognizes that the user has supplied content, and fills that content into the appropriate field. To change contacts if desired, the user has only to click the contact name to bring up a contact selector, or move the cursor to the contact field and speak a new name. In the future,

Fig. 3.4 Using Vlingo to send a text message



to make the software even more convenient and hands-free, we expect also to support the user speaking, “Contact: <name>.”

Finally, to send the text message, the user can click the Send button (highlighted by default) or can press the side key and say, “Send.”

3.7 Technology for Unconstrained Speech Input

A key enabler to create this style of speech input is to get rid of the need for application-constrained speech input. If we had to restrict users to particular words and phrases, we would not be able to provide the simple model for users we described above.

This of course presents a challenge since speech recognition on truly unconstrained input is not practical. We instead need to use modeling and adaptation techniques to achieve something close to this.

In particular, we have been successful in creating these interfaces using a set of techniques:

- **Hierarchical Language Model Based Speech Recognition:** We have replaced constrained grammars with very large vocabulary (millions of words) Hierarchical Language Models (HLMs). These HLMs are based on well-defined statistical models to predict what users are likely to say given the words they have spoken so far (“let’s meet at ___” is likely to be followed by something like “1 pm” or the name of a place). While there are no hard constraints, the models are able to take into account what this and other users have spoken in the particular text box in the particular application, and therefore improve with usage. Unlike previous generations of statistical language models, the new HLM technology scales to tasks requiring the modeling of millions of possible words (such as open web search, directory assistance, navigation, or other tasks where users are likely to use any of a very large number of words).

- **Adaptation:** In order to achieve high accuracy, we make use of significant amounts of automatic adaptation. In addition to adapting the HLMs, the system adapts to many user and application attributes such as learning the speech patterns of individuals and groups of users, learning new words, learning which words are more likely to be spoken into a particular application or by a particular user, learning pronunciations of words based on usage, and learning peoples' accents. The adaptation process can be seen in Fig. 3.5.
- **Server-side Processing:** The vlingo deployment architecture uses a small amount of software (about 50KB–90KB, depending on platform) on the mobile device for handling audio capture and the user interface. This client software communicates over the mobile data network to a set of servers which run the bulk of the speech processing. While this does make the solution dependent on the data network, it enables the use of the large amounts of CPU and memory resources needed for unconstrained speech recognition, and more importantly allows the adaptation described above to make use of usage data across all users.
- **Correction Interface:** While the techniques described above result in high accuracy speech recognition across users, there are still errors made by the speech recognizer. In addition, there will be situations where the user will prefer to enter text using the keypad on the phone (where they need privacy, are located in high noise environments, or when the speech system is unavailable due to lack of network coverage). Therefore, we have designed the user interface to allow the user to freely mix keypad entry and speech entry (at any time the user can either type on the keypad or push the “talk” button to speak), and to allow the user to correct the words coming back from the speech recognizer. Users can navigate through alternate choices from the speech recognizer (using the navigation buttons),

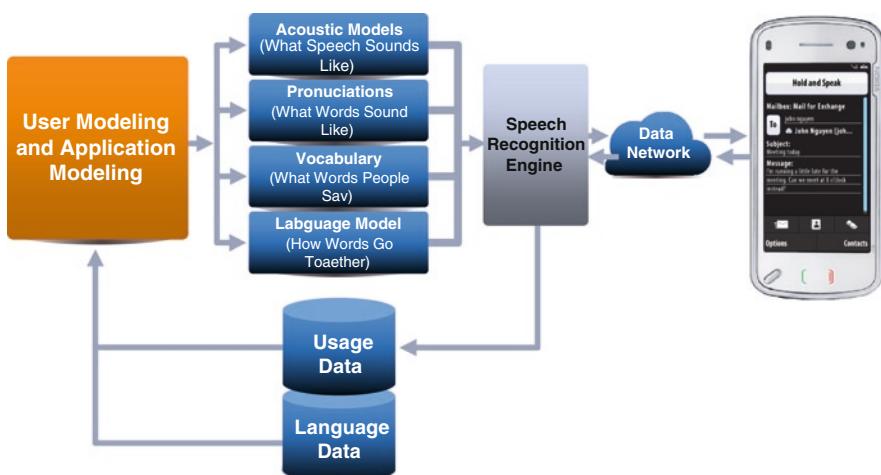


Fig. 3.5 Vlingo adaptation architecture. The core speech recognition engine is driven by a number of different models, each of which is adapted to improve its performance based on usage data

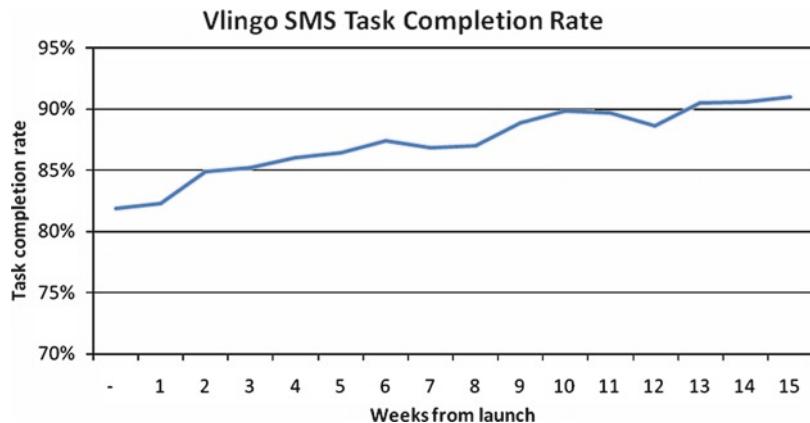


Fig. 3.6 Vlingo SMS task completion rate

can delete words or characters, can type or speak over any selected word, and can type or speak to insert or append new text wherever the cursor is positioned. We think this correction interface is the key to allowing users to feel confident that they can indeed efficiently enter any arbitrary text through the combination of speech and keypad entry.

As an example of the effects of adaptation, the chart above shows (Fig. 3.6) the progress in how often users who try to send an SMS with Vlingo complete the task. When Vlingo launched, even initial users experienced high success rates of 82%, which grew to over 90% over the subsequent 15 weeks. This significant improvement is due to a combination of accuracy gains from adapting to usage data, repeat usage which is more focused on real tasks instead of experimenting, and users learning to use the system more effectively.

3.8 Technology for Mapping to Actions

The other main technology component is to take word strings from users and map them to actions (such as in the case where the user is speaking a top-level input such as “send message to...”).

Our goal is to do this in a very broad way – allow the user to say whatever they want and to find some appropriate action to take based on this input. Because we want this to be broad, we also feel it likewise needs to be shallow. We feel it is reasonable for the speech interface to determine which application is best suited to handle the input, but that the applications are then the domain experts and that the speech interface should leave it up to them to interpret the input in some reasonable way for that application domain.

For this “intent modeling” we are also using statistical modeling techniques. For this component, we are developing statistical models which map input word strings

Vlingo BlackBerry Usage by Function

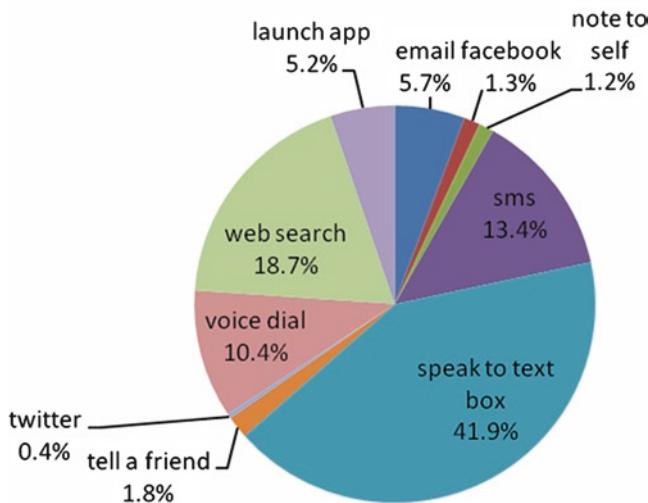


Fig. 3.7 Vlingo BlackBerry usage by function

to actions. We seed these models to a reasonable starting point, using knowledge of the domain and then adapt a better model for real input based on usage.

We also find that we can reduce the input variety by giving the users feedback on a “canonical” way of expressing top level routing input. The general form is “<application_or_action> <content>,” such as “web search restaurants in cambridge” or “navigate to 17 dunster street cambridge massachusetts.” We provide this feedback in help screens and in audio feedback to the user.

The combined effect of these approaches has led to successful deployments of these unconstrained speech interfaces. Not only are users able to achieve sufficient accuracy for various tasks, but they have come to view these interfaces as broadly applicable across various tasks. Figure 3.7 shows a snapshot of usage data from our deployment on Blackberry phones. The “speak to text box” usage is the case where users speak into an existing application (hence, using the speech interface as a keyboard into an existing application). The other usage types are where users speak at the top level of the phone to perform some specific function.

3.9 Usability Metrics and Results

It is becoming increasingly hard to find users who do not have mobile phones, and clearly everyone who works in the mobile domain is an active user. It is easy, therefore, to fall prey to the mistaken assumption that since each of us uses a mobile phone (or several mobile phones!), then all users must be like us and have the same goals. Perhaps shockingly for those in the industry, we find that the typical mobile user

simply does not share our level of investment in speech recognition software or our interest in the technical details. Our involvement in the industry necessarily creates a form of tunnel vision that requires the intervention of our users to, in a manner of speaking, save us from ourselves.

To maintain a single-pointed focus on meeting user goals over advancing technology for its own sake, we incorporate user research, data mining and usability testing into every major project release. For each key feature, we revisit the list of user personas, identifying what goals we will help our primary users achieve, what context they will be in when attempting to achieve those goals, and what elements are required to make the process easier, more efficient, and more satisfying.

Throughout the product lifecycle we draw on numerous tools from the field of user experience. Most notably:

- **During release definition phase:** usage data, surveys, interviews, and focus groups
- **During design and development phase:** iterative design, usability testing of paper prototypes, and live software
- **During quality assurance phase:** beta testing
- **Post-release:** usage data, surveys, and reviewing support incidents

For those readers who may be unfamiliar, we will briefly describe some of the key activities.

3.9.1 Usage Data

Because Vlingo is a network-based service, we have access to utterances spoken to Vlingo. To balance privacy with the research necessary to improve our application, we use abstract device IDs or device-stored cookies to discern utterances by user but have no way to identify any particular user. In other words, we can determine that some anonymous user spoke a particular combination of utterances and identify what was spoken, what the system heard, what action we took, and whether the user ultimately abandoned or completed the task.

Internal predictions, user requests, usability testing and even beta testing can tell us only so much about how real users will experience a product or feature during real situations. Through careful mining of usage data, we can evolve our intention engine – identifying new commands we should support for existing features as well as new features users expect to have voice activated. Usage data mining can also help us identify latent usability issues – specifically, tasks with low completion rates that merit further study.

3.9.2 Usability Testing

At Vlingo, we perform at least one, and often two, rounds of usability testing on each major feature. To date, we have conducted over 600 usability tests on various

aspects of the Vlingo software, studying everything from the most intricate details of the voice-enabled text box to high-level system evaluations of Vlingo on a particular platform. At Vlingo, usability testing can take several forms.

Early in the design phase, if there are multiple viable approaches, we may conduct what is known as A/B testing: here, we create paper mockups or prototypes of each approach and invite representative users to attempt to complete the same tasks with each of the prototypes. To prevent order effects, we rotate the order in which prototypes are presented: half the users start with prototype A, while the other half start with prototype B. We then compare the two prototypes according to user preference, task completion rates, error rates, and the severity of errors and usability issues encountered. Theoretically, A/B testing provides relatively objective data enabling a team to decide between the two concepts. In reality, however, it usually uncovers a third, more elegant design approach that incorporates the best of each alternative.

Later in the design phase, we perform more standard usability testing, in which 8–10 participants representative of our target user population are given a set of tasks to complete using a pre-release version of the software. We began by testing on our own hardware – recruiting users who have full keyboard BlackBerry devices, yet conducting the test on our own phones. We quickly learned that the mobile landscape is sufficiently unstructured so as to make testing more productive on users' devices. In this way, we uncover issues unique to particular carrier/device combinations, 3rd party applications, the vagaries of particular users' address books or setups, or in the case of BlackBerry Enterprise users, particular administrator policies, all of which we ordinarily might not learn until much later in the development or even the release process.

During the usability testing process, we provide the user with goals ("Send an email to a friend suggesting something to do tonight") rather than specific tasks. In this way, users fill in their own real-world expectations and content. During usability testing, we look for qualitative insights into how the system performs, where there are usability issues, and how they might be addressed. We also capture somewhat quantitative information such as task completion rates and user ratings of ease-of-use, usefulness, accuracy and speed. We say here "somewhat quantitative" because the number of participants is below that needed for statistical significance: at this point, we seek directional information rather than true statistical significance.

3.9.3 Beta Testing

We undertake two different types of beta testing: high-touch and low-touch. In high-touch testing, we provide users with particular tasks to complete and ask specific questions about their experience. This type of testing offers a deep dive into specific questions the team is grappling with or particular features identified as high-risk, but does not allow us to understand behavior "in the wild." As a result, any high-touch beta testing is also accompanied by extensive low-touch testing, in which participants are given the software, encouraged to use it for any tasks that they consider appropriate, and are sent a questionnaire 1 ½ to 2 weeks later.

Beta tests involve significantly larger sample sizes than do usability tests: for example, in a recent low-touch beta test of our *Vlingo Everywhere* feature that allows BlackBerry users to speak in order to fill in any text box, we included 500 beta testers, roughly evenly divided between new users and those who had used a previous version of the application. The sample size of low-touch beta tests allows us to employ more quantitative measures, as described below.

3.9.4 Usability Metrics and Findings

We use SUS, or the System Usability Scale, as a way to measure the usability of a product on its own, as well as to measure usability trends across releases. Developed by John Brooke of Digital Equipment Corporation in 1986, SUS presents ten statements to which users respond on a Likert scale from Strongly Disagree to Strongly Agree. From those ten responses, the instrument enables the practitioner to calculate a score of general system usability on a 100-point scale. SUS has been applied and shown to be reliable across diverse platforms and applications. In a 2004 study conducted at Fidelity by Tom Tullis and Jacqueline Stetson comparing several usability questionnaires, SUS showed the greatest reliability with the fewest number of users.

In their 2008 book *Measuring the User Experience*, Tom Tullis and Bill Albert describe a comprehensive review of published usability studies representing various applications and platforms, from which they determined that a score of over 80% is considered good usability (with a score of 77/100 representing the 75th percentile).

In light of Tullis and Albert’s findings, we have been pleased with Vlingo’s SUS scores. For example, in a survey of 162 Vlingo 2.0 BlackBerry beta users, the product earned a mean SUS score of 82.2 and a median rating of 88.8. This represented a positive trend over the ratings assigned to an earlier version of our software (Vlingo 1.1 earned mean: 77.9; median: 82.5).

In a recent user survey of 85 Vlingo users, we first asked about general attitudes and experiences with speech recognition systems. Not surprisingly, given the self-selecting nature of the survey population, respondents were excited about the promise of speech recognition – the ability to use their phones hands-free, the increased efficiency, and even the “cool” factor of using advanced technology. However, these users also felt that traditional speech recognition systems fell short of delivering on that promise, reporting that speech recognition systems are slow, do not understand them well enough and require them to “speak the way the system wants” rather than being able to speak naturally. In fact, when these users were shown a set of randomly ordered adjectives (half positive, half negative), and asked to choose the ones that applied to speech recognition systems they have used in the past, the only adjectives that received at least a third of the responses were: Frustrating, Error-Prone and Slow.

Later, we asked these users to choose the adjectives that applied to Vlingo. The responses were quite different: this time, the adjective that received at least a third of the responses were: Convenient, Useful, Cool, Simple, Easy, and Fun. The most popular negative adjective (Slow) was selected by only 15% of users.

Finally, when asked to rate the application, these users assigned Vlingo a 4.4 out of 5 for ease of use, 3.8 out of 5 for speech recognition accuracy, and 4.0 out of 5 overall rating.

3.10 Usage Data

As mentioned above, it is critical that the overall voice user interface design takes into account different types of users and usage scenarios. One example of this is shown in the Fig. 3.8, which illustrates that the behavior of initial users can be quite different than that of more experienced users. For example, initial users tend to clear results when faced with recognition issues, whereas expert users are much more likely to correct either by typing or selecting from alternate results, and also are more likely to compose complex messages by speaking multiple utterances.

User behavior also varies greatly depending on the type of device. The graph below (Fig. 3.9) shows several surprising results.

1. Although one would expect users of reduced-keyboard devices to type less than users of full-keyboard devices, the graph shows that users of reduced-keyboard device are actually more likely to correct by typing than users of other types of keyboards.

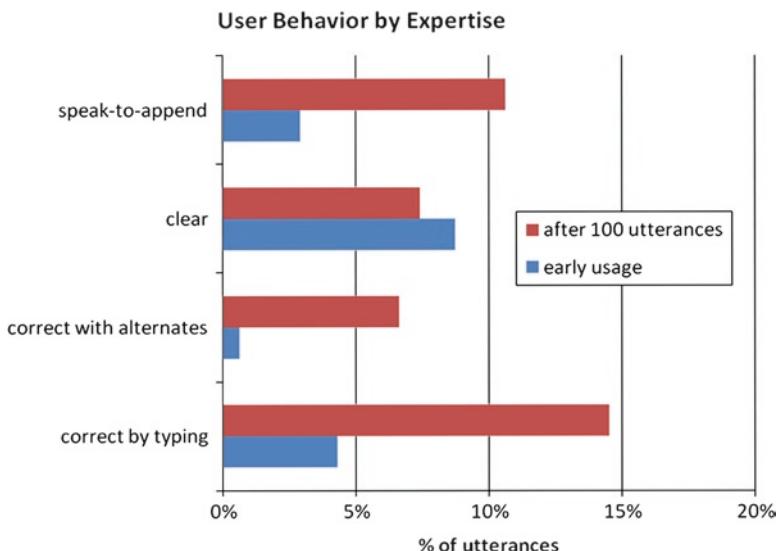


Fig. 3.8 User behavior by expertise

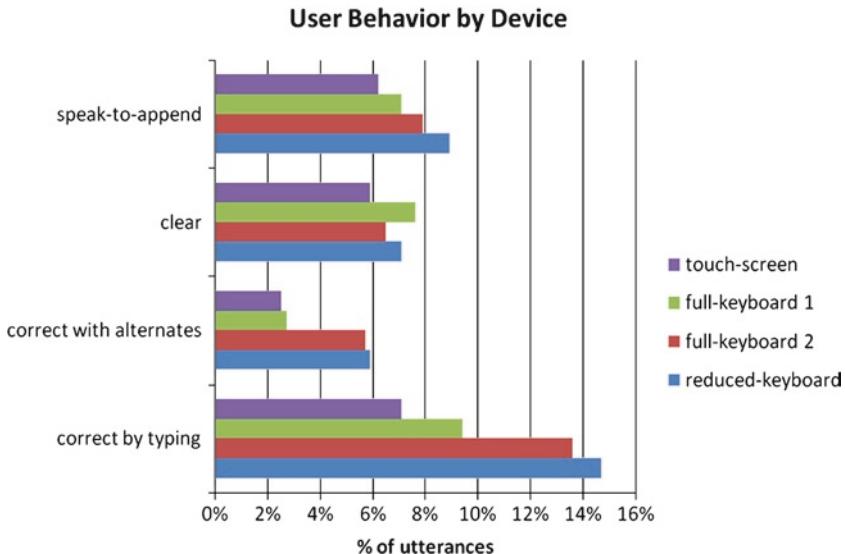


Fig. 3.9 User behavior by device

- Even for the same device physical profile, behavior can change significantly depending on the target users of each device. The full-keyboard devices shown as “full-keyboard 1” and “full-keyboard 2” are used differently. Usage of “full-keyboard 1” is much closer to the touchscreen case, whereas usage of “full-keyboard 2” is closer to that of reduced-keyboard. This is likely explained by the background that “full-keyboard 1” is a more consumer-focused device, while “full-keyboard 2” is a more business-focused device, with users who are more focused on getting tasks completed.
- Touch-screen correction is lower than that of other input modalities, most likely because users of that device are not as comfortable with typing and because tasks such as positioning the cursor on specific letter positions are more difficult on touch-screen devices.

By many measures, users interact differently with an automated system when compared with their interactions with another person. Figure 3.10 shows an example of this effect. When speaking to Vlingo, users accomplish most tasks with only a small number of words. A voice request to dictate an SMS is closer to a typed SMS than a spoken communication to another user, which would rarely be only 9 words. Social-network status updates to Facebook and Twitter are even shorter and also align well with the length of typed updates. Even for emails which tend to be more formal, users typically keep the length to well under 20 words.

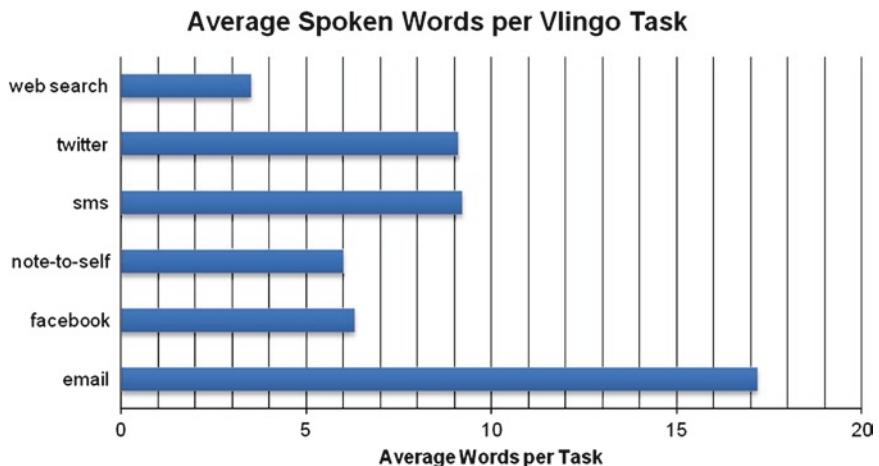


Fig. 3.10 Average spoken words per Vlingo task

3.11 Future of Mobile Speech Interfaces

There has been a tremendous amount of progress over the past few years. Just a few years ago, the state-of-the-art of mobile speech interfaces were mainly limited to very constrained device-based applications such as voice dialing. In addition to the systems that we are deploying, we now see speech interfaces in a number of point applications, including unconstrained speech recognition in voice search from multiple sources such as Microsoft's voice-enabled Bing, Google Search by Voice, and many others. We are also seeing dictation applications from major players such as Nuance. In addition, the latest Android phone released by Google includes a voice interface attached to the virtual keyboard, so any place where you can type, you can now speak.

But, there is still a long way to go to a truly ubiquitous multi-modal interface that works well across all applications and situations. The top-level application launching plus allowing speech input into any text field is the first step to this broad user interface. But, to fully make use of this functionality, applications are going to need to be designed with the speech interface in mind. While allowing speech into any text field does allow broad usage, if the application is designed to avoid text entry, it may not make good use of speech. For example, a navigation application may include separate fields for street number, street name, city name, state name, etc. and expect the user to type or select from dropdown lists in each of the fields. Forcing the user to scroll to each box and speak the input will work, but will not be nearly as convenient (or as safe) as just allowing them to say "navigate to 17 dunster street in cambridge".

We believe that once speech becomes part of the operating system of the phone that applications will evolve to take advantage of the changes in user behavior now

that they have the option for spoken input across applications. This is similar to what happened with touch-screen interfaces. While there were limited deployments of touch screen interfaces on various devices, the situation changed dramatically when Apple released the iPhone in 2007. By integrating touch as a key part of the operating system, they transformed the user experience not only on their own devices, but across the industry. In addition to prompting other mobile device makers to incorporate similar interfaces in their own devices, application developers started taking advantage of this interface in their application design to create a wide array of successful applications. We expect a similar transformation to take place over the next few years as speech is built into the operating systems of devices.

Once speech is built in as a key part of mobile phone operating systems (to achieve the goal of a broad interface), then we can truly make use of the potential of speech interfaces to allow much richer applications. Given the current constraints of text entry on mobile devices, mobile applications are designed to minimize the need for text entry – constraining the goals to what can be achieved with button and menu choices and small amounts of text entry (except of course for messaging applications which cannot avoid the need for text entry). Once there is a much easier and more natural way for people to interact with their mobile devices, applications can be much more ambitious about what they can do. In particular, we can start to provide much more open interfaces for people to perform various tasks.

Our overall goal is to allow people to say whatever they want, and then have their phone do the right thing across a broad set of possibilities. So, people should be able to say things like “schedule a meeting with me, Dave, and Joe tomorrow around lunchtime” and the phone should be able to interpret this, find the right applications which can handle it, and provide appropriate feedback to the user. Although this sort of thing is ambitious, it is an example of something that application developers and phone makers would not even contemplate without a speech interface (since users would never type in something like this on a small keyboard). Once we see ubiquitous deployments of mobile speech interfaces, we expect that there will indeed be applications developed with these more ambitious goals and that they will become more and more successful over time

Chapter 4

“Your Word is my Command”: Google Search by Voice: A Case Study

Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne,
Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope

Abstract An important goal at Google is to make spoken access ubiquitously available. Achieving ubiquity requires two things: *availability* (i.e., built into every possible interaction where speech input or output can make sense) and *performance* (i.e., works so well that the modality adds no friction to the interaction).

This chapter is a case study of the development of Google Search by Voice – a step toward our long-term vision of ubiquitous access. While the integration of speech input into Google search is a significant step toward more ubiquitous access, it has posed many problems in terms of the performance of core speech technologies and the design of effective user interfaces. Work is ongoing and no doubt the problems are far from solved. Nonetheless, we have at the minimum achieved a level of performance showing that usage of voice search is growing rapidly, and that many users do indeed become repeat users.

Keywords Mobile voice search • Speech recognition • Large-scale language models • Userinterface design • Unsupervised learning • Ubiquitous access • Smartphones • Cloud-basedcomputing • Mobile computing • Web search

4.1 Introduction

Using our voice to access information has been a part of science fiction ever since the days of Captain Kirk talking to the Star Trek computer. Today, with powerful smartphones and cloud-based computing, science fiction is becoming reality. In this chapter we give an overview of Google Search by Voice and our efforts to make speech input on mobile devices truly ubiquitous.

J. Schalkwyk (✉)
Senior Staff Engineer, Google, 1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA
e-mail: johans@google.com

The explosion in recent years of mobile devices, especially web-enabled smartphones, has resulted in new user expectations and needs. Some of these new expectations are about the nature of the services – e.g., new types of up-to-the-minute information (“where’s the closest parking spot?”) or communications (e.g., “update my facebook status to ‘seeking chocolate’”). There is also the growing expectation of ubiquitous availability. Users increasingly expect to have constant access to the information and services of the web. Given the nature of delivery devices (e.g., fit in your pocket or in your ear) and the increased range of usage scenarios (while driving, biking, walking down the street), speech technology has taken on new importance in accommodating user needs for ubiquitous mobile access – any time, any place, any usage scenario, as part of any type of activity.

A goal at Google is to make spoken access ubiquitously available. We would like to let the user choose – they should be able to take it for granted that spoken interaction is always an option. Achieving ubiquity requires two things: *availability* (i.e., built into every possible interaction where speech input or output can make sense) and *performance* (i.e., works so well that the modality adds no friction to the interaction).

This chapter is a case study of the development of Google Search by Voice – a step toward our long-term vision of ubiquitous access. While the integration of speech input into Google search is a significant step toward more ubiquitous access, it posed many problems in terms of the performance of core speech technologies and the design of effective user interfaces. Work is ongoing – the problems are far from solved. However, we have, at least, achieved a level of performance such that usage is growing rapidly, and many users become repeat users.

In this chapter we will present the research, development, and testing of a number of aspects of speech technology and user interface approaches that have helped improve performance and/or shed light on issues that will guide future research. There are two themes which underlie much of the technical approach we are taking: *delivery from the cloud* and *operating at large scale*.

Delivery from the cloud: Delivery of services from the cloud enables a number of advantages when developing new services and new technologies. In general, research and development at Google is conducted “in-vivo” – as part of actual services. This way, we benefit from an ongoing flow of real usage data. That data is valuable for guiding our research in the directions of most value to end-users, and supplying a steady flow of data for training systems. Given the statistical nature of modern speech recognition systems, this ongoing flow of data for training and testing is critical. Much of the work described later, including core technology development, user interface development, and user studies, depends critically on constant access to data from real usage.

Operating at scale: Mobile voice search is a challenging problem for many reasons – for example, vocabularies are huge, input is unpredictable, and noise conditions may vary tremendously because of the wide-ranging usage scenarios while mobile. Additionally, well known issues from earlier deployments of speech technology, such as dealing with dialectal variations, are compounded by the large scale nature of voice search.

Our thesis in handling these issues is that we can take advantage of the large amount of compute power, a rapidly growing volume of data, and the infrastructure available at Google to process more data and model more conditions than ever done before in the history of speech recognition. Therefore, many of the techniques and research directions described later are focused on building models at scale – i.e., models that can take advantage of huge amounts of data and the compute power to train and run them. Some of the approaches discussed will be methods for exploiting large amounts of data – for example with “unsupervised learning,” i.e., the ability to train models on all the data that comes in, without the need for human intervention for transcription or labeling. Another key set of issues involve the question of how to “grow” models as more data becomes available. In other words, given much more data, we can train richer models that better capture the complexities of speech. However, there remain many open questions about the most effective ways to do so.

In addition to taking advantage of the cloud and our ability to operate at large scale, we also take advantage of other recent technology advances. The maturing of powerful search engines provides a very effective way to give users what they want if we can recognize the words of their query. The recent emergence of widely used multimodal platforms (smartphones) provides both a powerful user interface capability and a delivery channel.

This chapter presents the approaches we have taken to deliver and optimize the performance of spoken search, both from the point of view of core technology and user interface design. In Sect. 2 we briefly describe the history of search by voice efforts at Google. Section 3 provides an in depth description of the technology employed at Google and the challenges we faced to make search by voice a reality. In Sect. 4 we explore the user interface design issues. Multimodal interfaces, combining speech and graphical elements, are very new, and there are many challenges to contend with as well as opportunities to exploit. Finally, in Sect. 5 we describe user studies based on our deployed applications.

4.2 History

4.2.1 *GOOG-411*

Searching for information by voice has been a part of our every day lives since long before the internet became prevalent. It was already the case 30 years ago that, if you needed information for a local business, the common approach was to dial directory assistance (411 in the US) and ask an operator for the telephone number.

800-GOOG-411 [2] is an automated system that uses speech recognition and web search to help people find and call businesses. Initially, this system followed the well known model of first prompting the user for the “city and state” followed by the desired business listing as depicted in Fig. 4.1.

| | |
|-----------------|--|
| GOOG411: | Calls recorded... Google! What city and state? |
| Caller: | <i>Palo Alto, California</i> |
| GOOG411: | What listing? |
| Caller: | <i>Patxis Chicago Pizza</i> |
| GOOG411: | Patxis Chicago Pizza, on Emerson Street. I'll connect you... |

Fig. 4.1 Early dialog for a GOOG-411 query

| | |
|-----------------|---|
| GOOG411: | Calls recorded... Google! Say the business, and the city and state. |
| Caller: | <i>Patxis Chicago Pizza in Palo Alto.</i> |
| GOOG411: | Patxis Chicago Pizza, on Emerson Street. I'll connect you... |

Fig. 4.2 Single shot dialog for a GOOG-411 query

This basic dialog has been ingrained in our minds since long before interactive voice response systems (IVR) replaced all or part of the live operator interaction.

Pre-IVR systems use “store-and-forward” technology that records the “city-and-state” the caller is requesting and then plays the city and state to the operator. This frees the operator from direct interaction with the user and results in substantial savings of human labor. Additionally, it constrains the search for businesses to the chosen city.

In 2008, we deployed a new version of GOOG-411 which allowed (and encouraged) the user to state their need in a single utterance rather than in sequential utterances that split apart the location and the business (Fig. 4.2). This was motivated by our desire to accommodate faster interactions as well as allow the user greater flexibility in how they describe their needs. This approach introduces new speech recognition challenges, given that we can no longer constrain the business listing language model to only those businesses in or near the chosen city. In [10] we investigated the effect of moving from a city conditional to nation wide language model that allows recognition of the business listing as well as the location in a single user response.

Moving from a two-step to a single-step dialog allowed for faster and arguably more natural user interactions. This, however, came at the price of increased recognition complexity, for the reasons described earlier. This was our first step moving from traditional directory assistance to more complex systems. The next step was to exploit new modalities.

4.2.2 *Google Maps for Mobile (GMM)*

Traditional directory assistance applications are limited to a single modality, using voice as both input and output. With the advent of smartphones with large screens and data connectivity, we could move to a multimodal user interface with speech or text as the input modality, and maps with super-imposed business listings as the output modality.

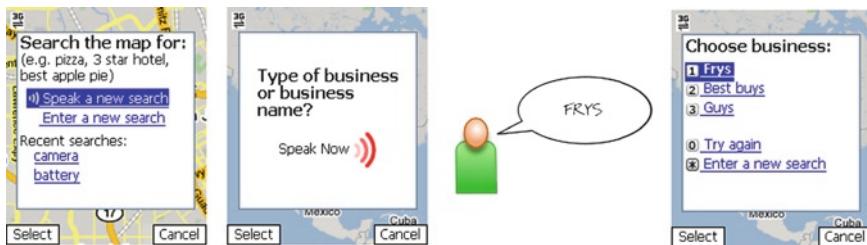


Fig. 4.3 Google maps for mobile, with voice interface

In March 2008 we introduced our first multimodal speech application for GMM. Figure 4.3 depicts a multimodal interface for directory assistance that we built on top of GMM.

A multimodal experience has some distinct advantages compared to the IVR (voice-only) system. First, the output modality can be visual rather than spoken, allowing much richer information flow. GMM can show the location of the business and other related information directly on a map. The contact information, address, and any other meta information about the business (such as ratings) can easily be displayed. A second major advantage relates to the time it takes the user to both search for and digest information. Due to the multimodality of the search experience, the total time spent is significantly less than the single input/output spoken modality of the IVR system. Finally, the cognitive load on the user is greatly reduced – the ephemeral nature of speech places significant cognitive demands on a user when the information communicated is lengthy or complex. These advantages enable a substantial improvement in the quality of interaction and quality of information one can provide compared to traditional IVR systems.

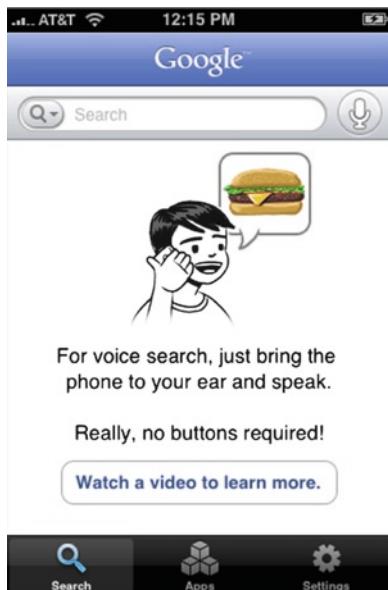
4.2.3 *Google Search by Voice*

Mobile web search is a rapidly growing area of interest. Internet-enabled smartphones account for an increasing share of the mobile devices sold throughout the world, and most models offer a web browsing experience that rivals desktop computers in display quality. Users are increasingly turning to their mobile devices when doing web searches, driving efforts to enhance the usability of web search on these devices.

Although mobile device usability has improved, typing search queries can still be cumbersome, error-prone, and even dangerous in some usage scenarios.

In November 2008 we introduced Google Mobile App (GMA) for iPhone (Fig. 4.4) that included a search by voice feature. GMA search by voice extended the paradigm of multimodal voice search from searching for businesses on maps to searching the entire world wide web. In the next few sections we discuss the technology behind these efforts and some lessons we have learned by analyzing data from our users.

Fig. 4.4 Google search by voice for iPhone



4.3 Technology

The goal of Google search by Voice is to recognize any spoken search query. Table 4.1 lists some example queries, hinting at the great diversity of inputs we must accommodate. Unlike GOOG-411, which is very domain-dependent, Google search by Voice must be capable of handling anything that Google search can handle. This makes it a considerably more challenging recognition problem, because the vocabulary and complexity of the queries is so large (more on this later in the language modeling Sect. 3.4).

Figure 4.5 depicts the basic system architecture of the recognizer behind Google search by Voice. For each key area of acoustic modeling and language modeling we will describe some of the challenges we faced as well as some of the solutions we have developed to address those unique challenges.

In Sect. 3.1 we will review some of the common metrics we use to evaluate the quality of the recognizer. In Sects. 3.2–3.4, we describe the algorithms and technologies used to build the recognizer for Google search by Voice.

4.3.1 Metrics

Choosing appropriate metrics to track the quality of the system is critical to success. The metrics drive our research directions as well as provide insight and guidance for solving specific problems and tuning system performance. We strive to find

Table 4.1 Example queries to Google search by voice

| Example query |
|--|
| Images of the grand canyon |
| What's the average weight of a rhinoceros |
| Map of san francisco |
| What time is it in bangalore |
| Weather scarsdale new york |
| Bank of america dot com |
| A T and T |
| Eighty-one walker road |
| Videos of obama state of the union address |
| Genetics of color blindness |

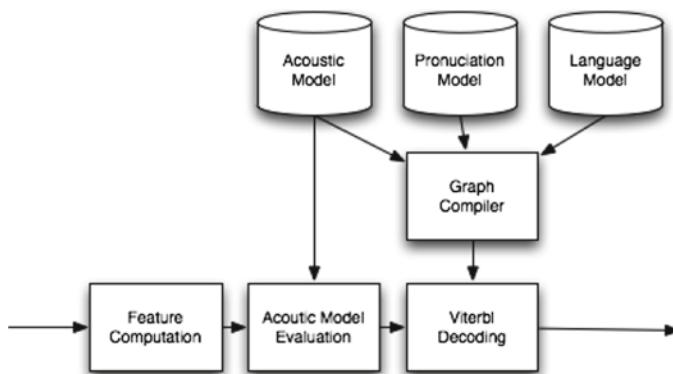


Fig. 4.5 Basic block diagram of a speech recognizer

metrics that illuminate the end-user experience, to make sure that we optimize the most important aspects and make effective tradeoffs. We also design metrics which can bring to light specific issues with the underlying technology. The metrics we use include:

1. Word Error Rate (WER):

The word error rate measures misrecognitions at the word level: it compares the words outputted by the recognizer to those the user really spoke. Every error (substitution, insertion, or deletion) is counted against the recognizer.

$$\text{WER} = \frac{\text{Number of Substitution} + \text{Insertions} + \text{Deletions}}{\text{Total number of words}}.$$

2. Semantic Quality (WebScore):

For Google search by Voice, individual word errors do not necessarily effect the final search results shown. For example, deleting function words like “in” or “of” generally do not change the search results. Similarly, misrecognition of the plural form of a word (missing “s”) would also not generally change the search results.

We, therefore, track the semantic quality of the recognizer (WebScore) by measuring how many times the search result as queried by the recognition hypothesis varies from the search result as queried by a human transcription. A query is considered correct if the web search result for the top hypothesis matches the web search result for the human transcription.

$$\text{WebScore} = \frac{\text{Number of correct search results}}{\text{Total number of spoken queries}}.$$

A better recognizer has a higher WebScore. The WebScore gives us a much clearer picture of what the user experiences when they search by voice. In all our research we tend to focus on optimizing this metric, rather than the more traditional WER metric defined earlier.

3. Perplexity (PPL):

Perplexity is, crudely speaking, a measure of the size of the set of words that can be recognized next, given the previously recognized words in the query.

The aim of the language model is to model the unknown probability distribution $p(x)$ of the word sequences in the language. Let q represent an n -gram model of the language trained on text data for the language. We can now evaluate the quality of our model q by asking how well it predicts a separate test sample x_1, x_2, \dots, x_N also drawn from p . The perplexity of the model q is defined as:

$$PPL = 2^{\sum_{i=1}^N \frac{1}{N} \log_2 q(x_i)}.$$

This gives us a rough measure of the quality of the language model. The lower the perplexity, the better the model is at predicting the next word.

4. Out-of-Vocabulary (OOV) Rate:

The out-of-vocabulary rate tracks the percentage of words spoken by the user that are not modeled by our language model. It is important to keep this number as low as possible. Any word spoken by our users that is not in our vocabulary will ultimately result in a recognition error. Furthermore, these recognition errors may also cause errors in surrounding words due to the subsequent poor predictions of the language model and acoustic misalignments.

5. Latency:

Latency is defined as the total time (in seconds) it takes to complete a search request by voice. More precisely, we define latency as the time from when the user finishes speaking until the search results appear on the screen. Many factors contribute to latency as perceived by the user: (a) the time it takes the system to detect end-of-speech, (b) the total time to recognize the spoken query, (c) the time to perform the web query, (d) the time to return the web search results back to the client over the network, and (e) the time it takes to render the search results in the browser of the users phone. Each of these factors are studied and optimized to provide a streamlined user experience.

4.3.2 Acoustic Modeling

Acoustic models provide an estimate for the likelihood of the observed features in a frame of speech given a particular phonetic context. The features are typically related to measurements of the spectral characteristics of a time-slice of speech. While individual recipes for training acoustic models vary in their structure and sequencing, the basic process involves aligning transcribed speech to states within an existing acoustic model, accumulating frames associated with each state, and re-estimating the probability distributions associated with the state, given the features observed in those frames. The details of these systems are extensive, but improving models typically includes getting training data that is strongly matched to the particular task and growing the numbers of parameters in the models to better characterize the observed distributions. Larger amounts of training data allow more parameters to be reliably estimated.

There are two levels of bootstrapping required. Once a starting corpus is collected, there are bootstrap training techniques to grow acoustic models starting with very simple models (i.e., single-Gaussian context-independent systems). But there is another bootstrapping problem at the level of the application definition. In order to collect ‘real data’ matched to users actually interacting with the system, we need an initial system with acoustic and language models. For Google search by Voice, we used GOOG-411 acoustic models together with a language model estimated from web query data. There is a balance to maintain in which the application needs to be compelling enough to attract users, but not so challenging from a recognition perspective that it makes too many errors and is no longer useful. Google makes it easy to push the boundaries of what might be possible while engaging as many users as possible – partly due to the fact that delivering services from the cloud enables us to rapidly iterate and release improved versions of systems.

Once we fielded the initial system, we started collecting data for training and testing. For labeling we have two choices: supervised labeling where we pay human transcribers to write what is heard in the utterances and unsupervised labeling where we rely on confidence metrics from the recognizer and other parts of the system together with the actions of the user to select utterances which we think the recognition result was likely to be correct. We started with supervised learning, aggressively transcribing data for training, and then migrated toward unsupervised learning as the traffic increased.

4.3.2.1 Accuracy of an Evolving System

The basic form of the acoustic models used are common in the literature. The experiments shown here all use 39-dimensional PLP-cepstral [5] coefficients together with online cepstral normalization, LDA (stacking 9 frames), and STC [3]. The acoustic models are triphone systems grown from decision trees, and use

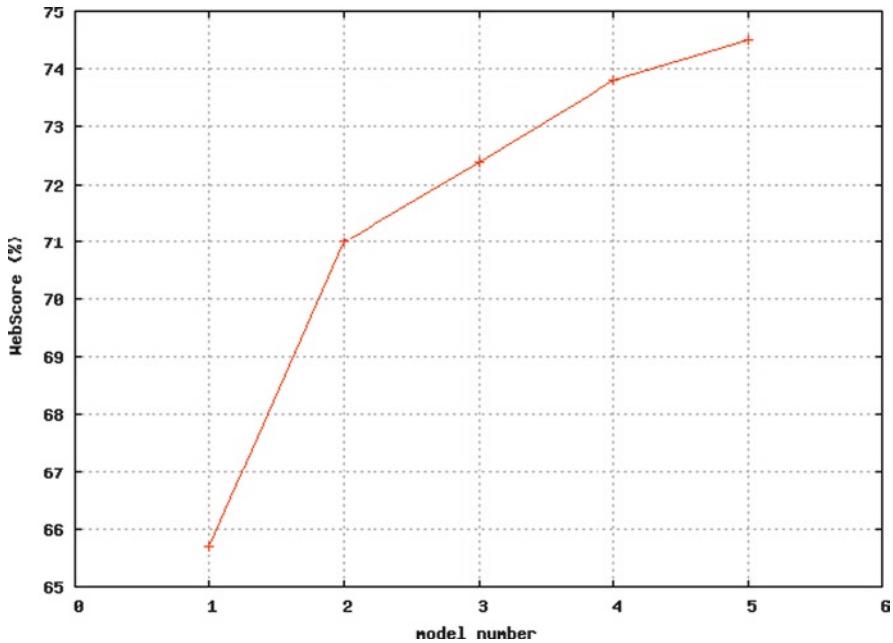


Fig. 4.6 WebScore evolution over time

GMMs with variable numbers of Gaussians per acoustic state. We optimize ML, MMI, and 'boosted'-MMI [8] objective functions in training.

Figure 4.6 shows the accuracy of the system on an off-line test set across various acoustic models developed in the first year of production. Each point on the x-axis represents a different acoustic model. These evaluations all use the same production language model (LM) estimated toward the end of the first year of deployment, but change the underlying acoustic model. The test set has 14K utterances and 46K words. The metric used here is WebScore, described earlier, which provides a measure of sentence-level semantic accuracy.

The first point on the graph shows the baseline performance of the system with mismatched GOOG-411 acoustic models. The second point, model 2, largely shows the impact of matching the acoustic models to the task using around 1 K h of transcribed data. For model 3, we doubled the training data and changed our models to use a variable number of Gaussians for each state. Model 4 includes boosted-MMI and adds around 5 K h of unsupervised data. Model 5 includes more supervised and unsupervised data, but this time sampled at 16 KHz.

Potential bugs in experiments make learning from negative results sketchy in speech recognition. When some technique does not improve things there is always the question of whether the implementation was wrong. Despite that, from our collection of positive and negative experiments we have seen a few general trends. The first is the expected result that adding more data helps, especially if we can keep increasing the model size. This is the basic engineering challenge in the field.

We are also seeing that most of the wins come from optimizations close to the final training stages. Particularly, once we moved to ‘elastic models’ that use different numbers of Gaussians for different acoustic states (based on the number of frames of data aligned with the state), we saw very little change with wide-ranging differences in decision tree structure. Similarly, with reasonably well-defined final models, optimizations of LDA and CI modeling stages have not led to obvious wins with the final models. Finally, our systems currently see a mix of 16 kHz and 8 kHz data. While we have seen improvements from modeling 16 kHz data directly (compared to modeling only the lower frequencies of the same 16 kHz data), so far we do better on both 16 kHz and 8 kHz tests by mixing all of our data and only using spectra from the first 4 kHz of the 16 kHz data. We expect this result to change as more traffic migrates to 16 kHz.

4.3.2.2 Next Challenges

The growing user base of voice search together with Google’s computational infrastructure provides a great opportunity to scale our acoustic models. The inter-related challenges include how and where to add acoustic parameters, what objective functions to optimize during training, how to find the optimal acoustic modeling size for a given amount of data, how to field a realtime service with increasingly large acoustic models, and how to get reliable labels for exponentially increasing amounts of data. Early experiments in these directions suggest that the optimal model size is linked to the objective function: the best MMI models may come from ML models that are smaller than the best ML models; that MMI objective functions may scale well with increasing unsupervised data; that speaker clustering techniques may show promise for exploiting increasing amounts of data; and that combinations of multicore decoding, optimizations of Gaussian selection in acoustic scoring, and multipass recognition provide suitable paths for increasing the scale of acoustic models in realtime systems.

4.3.3 *Text Normalization*

We use written queries to google.com in order to bootstrap our language model for Google search by Voice. The large pool of available queries allows us to create rich models. However, we must transform written form into spoken form prior to training. This section discusses our approach to text normalization, i.e., the approach by which we perform that transformation.

Written queries contain a fair number of cases which require special attention to convert to spoken form. Analyzing the top million vocabulary items before text normalization we see approximately 20% URLs and 20+% numeric items in the query stream. Without careful attention to text normalization the vocabulary of the system will grow substantially.

We adopt a finite state [1] approach to text normalization. Let $T(\text{written})$ be an acceptor that represents the written query. Conceptually, the spoken form is computed as follows:

$$T(\text{spoken}) = \text{bestpath}(T(\text{written})^{\circ}N(\text{spoken})),$$

where $N(\text{spoken})$ represents the transduction from written to spoken form. Note that composition with $N(\text{spoken})$ might introduce multiple alternate spoken representations of the input text. For the purpose of computing n-grams for spoken language modeling of queries we use the ‘bestpath’ operation to select a single most likely interpretation.

4.3.3.1 Text Normalization Transducers

The text normalization is run in multiple phases. Figure 4.7 depicts the text normalization process.

In the first step we annotate the data. In this phase we classify parts (sub-strings) of queries into a set of known categories (e.g., time, date, url, and location).

Once the query is annotated, it is possible to perform context-aware normalization on the substrings. Each category has a corresponding text normalization transducer $N_{\text{cat}}(\text{spoken})$ that is used to normalize the substring. Depending on the category we either use rule-based approaches or a statistical approach to construct the text normalization transducer.

For numeric categories like date, time, and numbers it is easy enough to describe $N(\text{spoken})$ using context dependent rewrite rules.

The large number of URLs contained in web queries poses some challenging problems. There is an interesting intersection between text normalization of URL

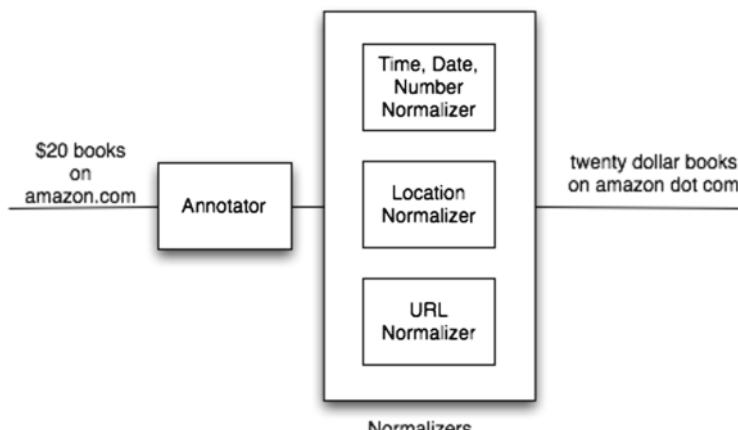


Fig. 4.7 Category/Context specific text normalization

queries and segmentation of text for languages like Japanese and Mandarin Chinese. Both require segmenting the text into its corresponding word constituents [9]. For example, one reads the URL cancercentersofamerica.com as “cancer centers of america dot com”. For the URL normalizer $N_{\text{url}}(\text{spoken})$ we train a statistical word decomposer that segments the string.

4.3.4 Large-scale Language Modeling

In recent years language modeling has witnessed a shift from advances in core modeling techniques (in particular, various n-gram smoothing algorithms) to a focus on scalability. The main driver behind this shift is the availability of significantly larger amounts of training data that are relevant to automatic speech recognition problems.

In the following section we describe a series of experiments primarily designed to understand the properties of scale and how that relates to building a language model for modeling spoken queries to google.com. A typical Voice Search language model is trained on over 230 billion words. The size of this data set presents unique challenges as well as new opportunities for improved language modeling.

Ideally, one would build a language model on spoken queries. As mentioned earlier, to bootstrap we start from written queries (typed) to google.com. After text normalization we select the top 1 million words. This results in an out-of-vocabulary (OOV) rate of 0.57%. Table 4.2 depicts the performance of the language model on unseen query data (10 K) when using Katz smoothing [7].

The first language model (LM) which has approximately 15 million n -grams is used for constructing the first pass recognition network. Note this language model requires aggressive pruning (to about 0.1% of its unpruned size). The perplexity hit taken by pruning the LM is significant – 50% relative. Similarly, the 3-gram hit ratio is halved.

The question we wanted to ask is how does the size of the language model effect the performance of the system. Are these huge numbers of n -grams that we derive from the query data important?

Figure 4.8 depicts the WER and WebScore for a series of language models increasing in size from 15 million n -grams up to 2 billion n -grams. As the size of

Table 4.2 Typical Google Voicesearch LM, Katz smoothing: the LM is trained on 230 billion words using a vocabulary of 1 million words, achieving out-of-vocabulary rate of 0.57% on test data

| Order | No. n-grams | Pruning | PPL | n-gram hit-ratios |
|-------|-------------|---------------------|-----|-------------------|
| 3 | 15M | Entropy (Stolcke) | 190 | 47/93/100 |
| 3 | 7.7B | None | 132 | 97/99/100 |
| 5 | 12.7B | Cut-off (1-1-2-2-2) | 108 | 77/88/97/99/100 |

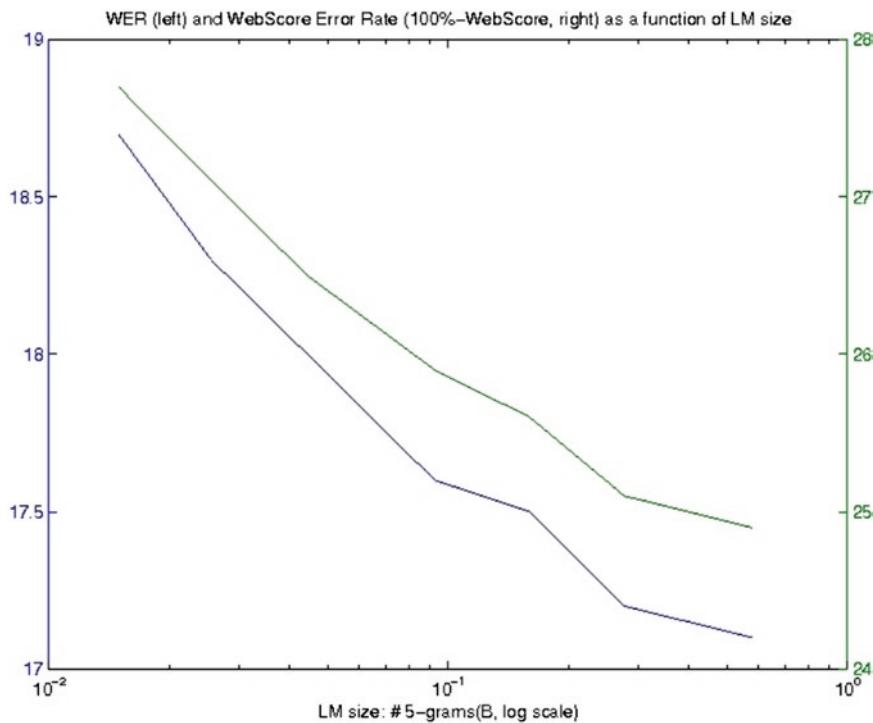


Fig. 4.8 Word Error (WER) and WebScore as a function of language model size

the language model increases we see a substantial reduction in both the word error rate and associated WebScore [4].

Figure 4.9 depicts the WER and the Perplexity for the same set of language models. We find a strong correlation between the perplexity of the language model and the word error rate. In general perplexity has been a poor predictor of the corresponding word error, so these results were rather surprising.

4.3.4.1 Locale Matters

We ran some experiments to examine the effect of locale on language model quality. We built locale specific English language models using training data from prior to September 2008 across three English locales: USA, Britain, and Australia. The test data consisted of 10k queries for each locale sampled randomly from September to December 2008.

Tables 4.3–4.5 show the results. The dependence on locale is surprisingly strong: using an LM on out-of-locale test data doubles the OOV rate and perplexity.

We have also build a combined model by pooling all data, with the results shown on the last row of Table 4.5.

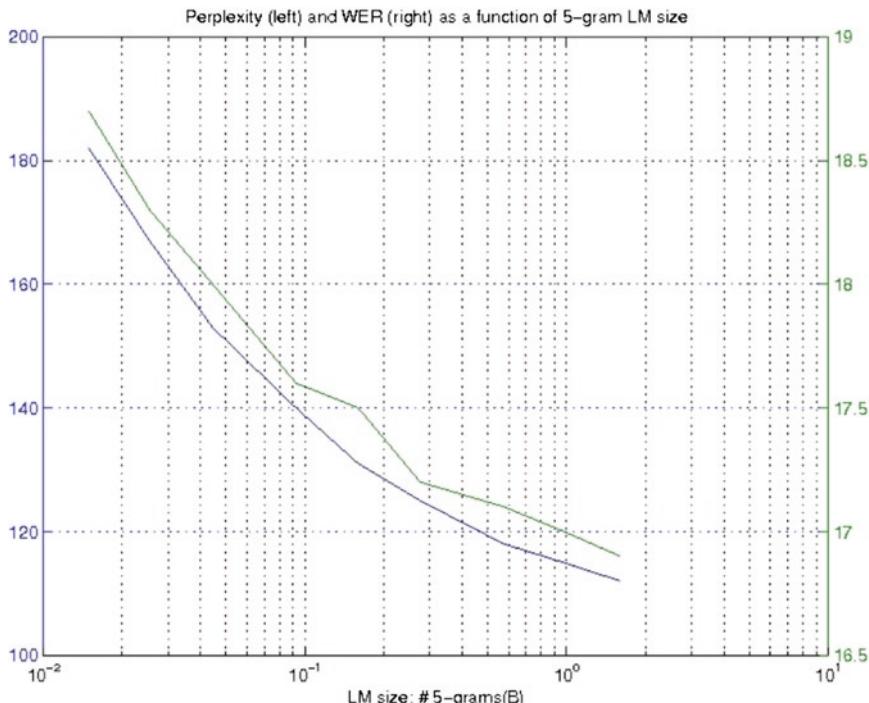


Fig. 4.9 Word error rate (WER) and perplexity as a function of language model size

Table 4.3 Out of vocabulary rate: locale specific vocabulary halves the OOV rate

| Training locale | Test locale | | |
|--------------------|-------------|-----|-----|
| | USA | GBR | AUS |
| USA | 0.7 | 1.3 | 1.6 |
| GBR | 1.3 | 0.7 | 1.3 |
| AUS | 1.3 | 1.1 | 0.7 |

Table 4.4 Perplexity of unpruned LM: locale specific LM halves the PPL of the unpruned LM

| Training locale | Test locale | | |
|--------------------|-------------|-----|-----|
| | USA | GBR | AUS |
| USA | 132 | 234 | 251 |
| GBR | 260 | 110 | 224 |
| AUS | 276 | 210 | 124 |

Table 4.5 Perplexity of pruned LM: locale specific LM halves the PPL of the unpruned LM

| Training locale | Test locale | | |
|--------------------|-------------|-----|-----|
| | USA | GBR | AUS |
| USA | 210 | 369 | 412 |
| GBR | 442 | 150 | 342 |
| AUS | 422 | 293 | 171 |
| combined | 227 | 210 | 271 |

Pooling all data is suboptimal

Combining the data negatively impacts all locales. The farther the locale from USA (as seen on the first line, GBR is closer to USA than AUS), the more negative the impact of clumping all the data together, relative to using only the data from that given locale.

In summary, we find that locale-matched training data resulted in higher quality language models for the three English locales tested.

4.4 User Interface

“Multimodal” features, like Google Search by Voice, provide a highly flexible and data-rich alternative to the voice-only telephone applications that preceded them. After all, they take advantage of the best aspects of both speech and graphical modalities. However, despite their benefits, multimodal applications represent largely uncharted territory in terms of user interface design. Consequently, there are many aspects that will need refinement or redesign. The good news is that, as more user data is gathered, we are gaining a much better understanding of the issues. What’s more, as more designers and developers try their hand at this type of interface this knowledge will grow even faster. In this section, we describe just a few of the unique characteristics that make multimodal applications both appealing to users as well as challenging for designers. For some of these challenges, we present viable solutions based on user data. For others, we describe ongoing experimentation that will ultimately lead to a better user experience.

4.4.1 *Advantages of Multimodal User Interfaces*

4.4.1.1 Spoken Input vs. Output

While speech is both convenient and effective as an input method, especially as an alternative to typing on tiny mobile keyboards, spoken output is very limited given its sequential nature. Consider the following examples from GOOG-411. The first involves a search for a specific restaurant named “Patxi’s Chicago Pizza” while the second shows a search for a common restaurant category, namely “pizza.”

As shown in Fig. 4.10, GOOG-411 handles specific name queries very efficiently, quickly connecting the caller to the business usually in about 30 s. However, when a caller specifies the business category as opposed to a specific name, as in Fig. 4.11, it takes more than a minute just to hear the first set of choices. If the caller chooses to further “browse” the list and perhaps listen to the details of one or two choices, the call time will be doubled. If it goes this far, however, there is a good chance the user will hang up without making a selection. It takes a great deal of time and concentration to process spoken information, and most user’s pain threshold is fairly low. While not conclusive, the GOOG-411 data supports this, as specific business name queries outnumber category searches more than five to one.

| | |
|-----------------|--|
| GOOG411: | Calls recorded... Google! Say the business and the city and state. |
| Caller: | <i>Patxi's Chicago Pizza in Palo Alto.</i> |
| GOOG411: | Patxi's Chicago Pizza, on Emerson Street. I'll connect you... |

Fig. 4.10 Specific business search with GOOG-411

| | |
|-----------------|--|
| GOOG411: | Calls recorded... Google! Say the business and the city and state. |
| Caller: | <i>Pizza in Palo Alto.</i> |
| GOOG411: | Pizza in Palo Alto... Top eight results: Number 1: Patxi's Chicago Pizza, on Emerson Street To select number one, press 1 or say "number one". Number 2: Pizza My Heart, on University Avenue. Number 3: Pizza Chicago, on El Camino Real. [...] Number 8: Spot a Pizza Place: Alma-Hamilton, on Hamilton Avenue |

Fig. 4.11 Business category search with GOOG-411

4.4.1.2 A Picture Paints a Thousand Words

Now consider the screens in Fig. 4.12 which show the results displayed for the same “Pizza in Palo Alto” query using Google’s voice search feature on Android. Not only does the user receive more information but also the graphical display allows much of it to be processed in parallel, saving a great deal of time.

The screen on the left shows the initial page displayed after recognition is complete, which includes the recognition result (pizza in palo alto) as well as the “n-best alternatives” (additional hypotheses from the recognizer) which are viewable by tapping on the phrase to display a drop-down list (note the down arrow on the right-hand side of the text field). The user can initiate a new search either by voice or by typing. As shown, the first three results are displayed in the browser, but tapping on “Map all results” delivers the full set of results in Google Maps, as shown on the right. The maps interface shows the relative location of each listing as well as the user’s contacts (note the blue box in the upper right-hand corner). Tapping the business name above the map pin provides more details.

4.4.1.3 Flexibility and User Control

Another general advantage of mobile voice search is the flexibility and control it affords users.

Unlike with voice-only applications, which prompt users for what to say and how to say it, mobile voice search is completely user initiated. That is, the user decides what to say, when to say it, and how to say it. There is no penalty for

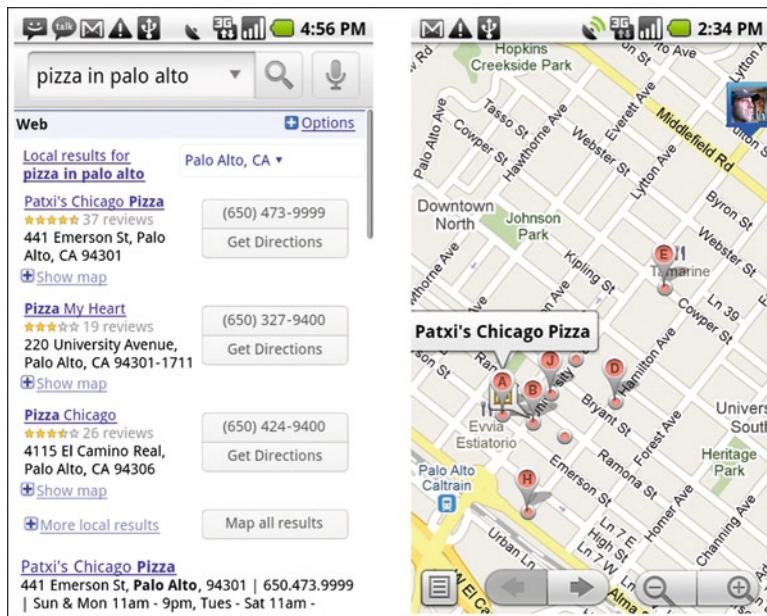


Fig. 4.12 Category search using Google search by voice

starting over or modifying a search. There is no chance of an accidental “hang-up” due to subsequent recognition errors or timeouts. In other words, it’s a far cry from the predetermined dialog flows of voice-only applications.

As we discussed earlier, spoken output can be hard to process, but given their flexibility, multimodal applications can still provide spoken output when it’s convenient. Consider queries like “Weather in Palo Alto California,” “Flight status of United Airlines 900,” “Local time in Bangalore,” and “Fifty pounds in US dollars.” These types of queries have short answers, exactly the kind suited for spoken output, especially in eyes-busy contexts.

Still, the flexibility associated with multimodal applications turns out to be a double-edged sword. More user control and choices also leads to more potential distractions. The application must still make it clear what the user can say in terms of available features. For example, in addition to web search, Google’s Android platform also includes speech shortcuts for its maps navigation feature, e.g., “Navigate to the Golden Gate Bridge,” as well as voice dialing shortcuts such as “Call Tim Jones.” More fundamental is making sure users know how to use the speech recognition feature in the first place given all the features available. Designers are faced with a series of hard questions: How should voice search be triggered? Should it be a button? A gesture? Both? What kind of button? Should it be held and released? Tapped once? Tapped twice? What kind of feedback should be displayed? Should it include audio? We address these and other questions in the subsections that follow.

4.4.2 Challenges in Multimodal Interface Design

4.4.2.1 Capturing the Utterance: Buttons, Actions, and Feedback

Capturing clear and complete user utterances is of paramount importance to any speech application. However, even if everything is done to ensure that the signal is clean and the microphone is working properly, there are factors in the user interface itself which will affect the interaction.

On the face of it, pressing a button to initiate speech seems pretty simple. But once you consider the types of buttons available on mobile devices as well as the actions possible for each type of button, and further the size of the button and where it's placed on the screen, things become more complex. Google uses different strategies depending on the device.

Devices running Android have the microphone button on the right-hand side of the search box typically located at the top of the home touch screen. This is similar to the button on Google Mobile App (GMA) for the iPhone, which also uses a touch screen. Both are shown in Fig. 4.13.

As shown earlier, both microphone buttons are relatively small, which raises the obvious question as to whether a bigger button would make it easier for users to trigger voice search or perhaps users would more easily discover the feature in the first place. Alternatively, the button could remain the same size but with a larger target area to trigger the action. This is currently the case in the GMA interface,

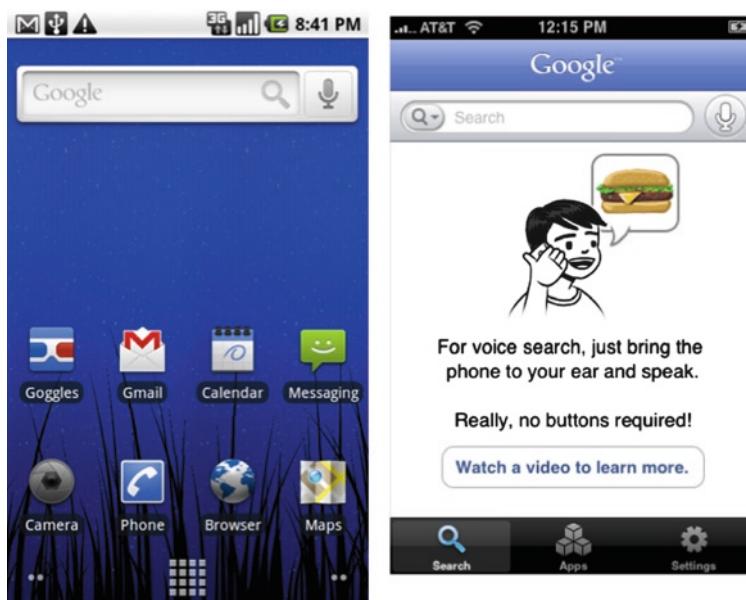


Fig. 4.13 Android nexus one and Google mobile app (GMA) on iPhone

shown on the right. So far, there is no evidence that the larger target area is making it any easier to trigger as compared to the Android button.

Then there is the placement of the button. In the examples shown earlier, the upper right-hand corner location may not be a factor when the user holds the phone with one hand and presses the button with the other. However, some users prefer to initiate speech with one hand. In this case, it may make a difference whether the user is right or left handed. Other such ergonomics-based suggestions have been proposed such as locating a larger button across the bottom of the screen so that users can hold the phone in one hand and more easily press the button with their thumb.

It should also be pointed out that there is a physical “search” key on all Android phones. A regular press (one tap) simply brings up the search widget from any context (i.e., no matter which app the users has open). However, long-pressing this button (holding it down for a second or so) brings up voice search. The long press is a common feature for Android as it is used in many contexts, that is, not just on physical buttons but on the touch screen itself. Note that this is not the same as the hold-and-speak, walkie-talkie action which is used for the BlackBerry and S60 versions of GMA, which we discuss later.

4.4.2.2 Button Actions

While most mobile speech apps require the user to press a button to initiate recording, only some require the user to manually stop the recording after speaking by pressing it again, or pressing another button. In the examples discussed in Fig. 4.2 above, both applications make use of an “endpointer,” which is software that automatically determines when the speaker’s utterance is complete (i.e., it finds the “end point”). This is the same strategy used in most speech-based telephone applications. While endpointers may be convenient for mobile speech, they seem to be better suited for applications like web search or voice commands in which the input is shorter, generally one phrase. This is because silence is a primary factor used to determine the end point of the utterance. In this way, applications that must tolerate longer periods of silence between phrases as in dictation or singing often require the user to tap the button once to begin and then a second time to manually end recording.

Another way to manually endpoint is to press and hold the button while speaking. This is based on the “walkie talkie” model. GMA employs this strategy on platforms with physical buttons, namely the BlackBerry as well as S60 platform phones. While the press-and-hold strategy seems intuitive and certainly has its fans, a common problem is the tendency to release the button before finishing the utterance. This premature endpointing in turn causes the utterance to be truncated, usually resulting in misrecognition.

4.4.2.3 Gesture-Based Speech Triggers

Putting buttons aside for the moment, gesture-based triggers for initiating speech are another strategy which has been implemented in the iPhone version of GMA, as shown on the right-hand screen in Fig. 4.13 above.

As the home screen hint says, voice search will trigger without a button press when the user simply raises the phone to his or her ear as this type of movement is detected by tapping into the phone’s accelerometer. While it turns out that many users like this feature (fully one third of voice searches on GMA for iPhone are triggered by this gesture), others still do not realize it exists despite the rather explicit hint shown on the splash screen. A Google internal study also showed that some users, while aware of the feature, prefer to keep their eyes on the screen at all times, something that is not possible when using this gesture.

4.4.2.4 Feedback

Even when users understand how to initiate speech, subsequent user interface feedback plays an important role. For example, in an early prelaunch design of voice search for GMA for iPhone, the word “listening” was used on the screen that appeared after the user pressed the microphone button. The designers assumed users would understand that “listening” clearly indicated that it was time for the user to speak. However, in several cases, the participants intently watched the “listening” screen but said nothing. When asked what the application might be doing, the users responded that device was “listening” to somehow calibrate the ambient noise-making sure noise levels were right. As a result, the application took a more direct approach. As shown in Fig. 4.14, all of Google’s mobile voice features begin with “Speak now.” In addition, they give clear feedback on the level of the speaker’s voice indicated in the Android case below by the microphone filling up as the level increases.

Making sure the recognizer at least starts with a clean and complete recording of what the user actually said is key for any speech application. As we have seen, this is far from automatic and many strategies are currently in use. It may be that some are equally effective and address different user preferences. However, we are also likely to discover that some are simply more effective.

4.4.3 Correction: Displaying Alternative Recognition Hypotheses

4.4.3.1 The N-Best List

Speech recognition is not perfect and designing speech-based applications requires paying special attention to these inevitable errors. One important design puzzle involves making what is referred to as the “n-best list” more accessible and visible to users. This is a list of alternative recognition hypotheses returned by the recognizer. For example, suppose the user says “Holy day in South America.” The recognizer may return “holiday inn south america” as the top hypothesis but include what the user actually said in the list. It may also be the case that what the user said is not in the list but there are alternatives that are sufficiently related. In these scenarios, making sure the n-best is easily accessible saves

Fig. 4.14 Feedback during speech input



the user the frustration of having to respeak the utterance and at the same time fosters a more positive impression of the application in general.

4.4.3.2 Design Strategies

Figure 4.15 later shows how Android displays the n-best list. A similar strategy is used in GMA for iPhone as well as for the BlackBerry and S60 platform.

As shown on the left, only the top hypothesis is displayed on the results screen. The down-arrow indicator is used to bring attention to the n-best list which is displayed if the user taps anywhere inside the text field. The right-hand screen shows the list displayed. As it turns out, this design is not effective as we'd like as only a small percentage of users are tapping on the phrase to reveal the list even when it would be helpful (i.e., when it contains the correct alternative or a related phrase).

4.4.3.3 Possible Solutions

There are several reasons for this. It may be that the drop down list is not obvious and we need a more prominent hint. Or it may be that users are aware of the drop down but are not sure what the list is for. That is, it could be that users do not realize

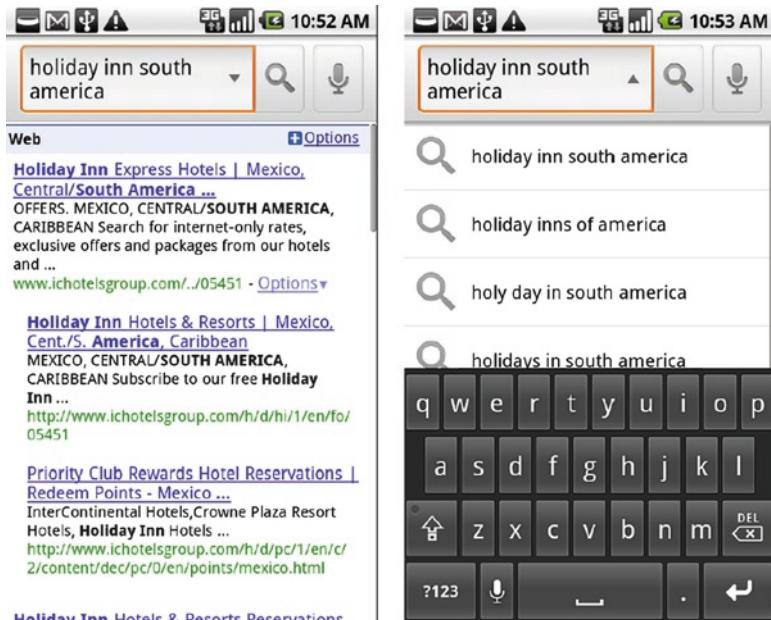


Fig. 4.15 Displaying n-best on Android

that tapping on the alternative would initiate a new search. It could also be that users do not find it's worth the trouble to tap the list just to see if the alternative is there and decide instead that they might as well just respeak the phrase.

Possible solutions will only emerge as we experiment with alternative designs. One idea is to make the n-best list more prominent by displaying a pop-up hint or even flashing the list for a second or two as the results load. However, we must also make sure not to burden users when the list is irrelevant either because the correct alternative is not there or because the top hypothesis is already correct. In general speed is king for mobile user interfaces and we try to do as little as possible to get in the way of displaying the results. This problem of relevance might be solved by taking advantage of the confidence score returned with the results. That is, the recognizer returns a score roughly indicating “how sure” it is the phrases it returns are what the user said. In this way, the design could more aggressively draw attention to the list when the confidence is lower and otherwise leave it as is otherwise. This experiment is in fact underway now but we’ll have to wait and see if it addresses all the user interface factors at play.

The problem of correction becomes even more complex when designing for dictation interfaces, where the correction of one phrase might affect the compatibility of preceding or subsequent phrases, for example. But dictation goes beyond the scope of this discussion.

4.4.4 *Beyond Search*

4.4.4.1 Nonsearch Voice Commands

Subsequent releases of the Android platform included new voice shortcuts using the same interface they had used for search (i.e., the same mic button and initial dialog screens). For example “Navigate to the Golden Gate Bridge” jumps straight to the Google Maps navigation feature and begins giving directions as though the user had tapped through and entered the destination by hand. Other commands like “Call John Smith at home” or “Map of Chelsea, Manhattan” likewise provide quick and easy ways to access embedded application functions just by speaking a phrase.

4.4.4.2 Challenges and Possible Solutions

This new functionality comes with a price, however. To quickly launch the features, particular phrases like “navigate to,” “map of,” “directions to,” “call,” etc. were mapped to each shortcut. However, it’s well known that users tend to paraphrase when faced with formulating the command on the fly especially if the targeted phrase is something unfamiliar or that does not match their own language patterns. For example, just because I once learned that the phrase “navigate to” will trigger the Google Maps feature in one step does not mean I will remember that exact phrase when I need it. In fact, I am likely to substitute the phrase with a synonymous phrase like “Take me to” or “Drive to.”

There are short- and long-term solutions for this. First, similar to the n-best list situation, contextually significant and visually prominent hints can help a great deal to remind users what the working phrases are. Subsequent designs for these features on Android will in fact include them. However, does this go far enough?

4.4.4.3 Predicting User Intent

Rather than requiring users to memorize specific phrases, a better solution would be for users to choose their own shortcut phrases. That is, they could say what they wanted and it would “just work.” Of course, this is easier said than done. The linguistic possibilities are endless and complex semantic parsing capabilities would be required even to begin to return reasonable candidates for what the user might have said. What’s more, unlike search, in this case you would be combining possible results for very different actions. “Call of the Wild” is a search while “Call Owen Wilde” is a contact dialing action, yet the two sound very similar. At the very least the application would need to display disambiguation lists much more often than it currently does so that the user could choose the option that he or she intended (if it’s there) and reinforce the feedback loop for better results. However, this would add an extra step before results could be displayed or the action carried out.

Automatically, knowing what users mean based on speech input is clearly a longer term project. However, from a user’s perspective it is not likely to be thought of as very different from what is currently offered in mobile apps. Think of Google Search by Voice: Users already say whatever phrase they want and are given a list of choices, often with the one they wanted listed right at the top.

4.5 User Studies

What are people looking for when they are mobile? What factors influence them to choose to search by voice or type? What factors contribute to user satisfaction? How do we maintain and grow our user base? How can speech make information access easier? In this section we explore these questions based on analysis of live data. We discuss the behavior of our users and how these impact our decisions about technology and user interfaces.

4.5.1 *What do Users Choose to Speak?*

In this section, we discuss the search patterns of voice search users. We investigate the use cases for search by voice, and how they differ from other methods of communicating a query to a search engine. We address these questions empirically, by looking at the server logs across various search platforms and input modalities.

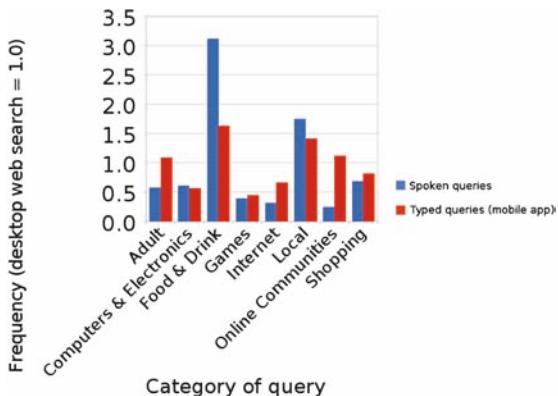
Our data set consists of the search logs for all users of GMA on all mobile platforms in the United States who issued queries during a 4-week (28-day) period during the summer of 2009. For a baseline, we also analyze search queries to the “desktop” (i.e., nonmobile) version of google.com.

In general we find the distribution of voice search queries to be surprisingly similar to the distributions for both mobile web search and desktop queries, but there are a number of interesting differences.

4.5.1.1 By Topic

We aggregated a recent month of our search server logs by the broad topic of the query using an automatic classification scheme described in Kamvar and Baluja [6]. The chart later illustrates the relative difference between spoken and typed queries for eight popular categories. Each bar is normalized so that 1.0 represents the frequency of that category for desktop websearch (Fig. 4.16).

Queries in the “Local” category are those whose results have a regional emphasis. They include queries for business listings (e.g., “Starbucks”) but can also include places (e.g., “Lake George”) or properties relating to a place

Fig. 4.16 Category of query

(“weather Holmdel NJ,” “best gas prices”). Food & Drink queries are self-descriptive and are often queries for major food chains (e.g., “Starbucks”), or genres of food & drink (e.g., “tuna fish,” “Mexican food”). Both of these query types likely relate to a user’s location, even if there is no location specified in the query (this facilitated by the My Location feature which will automatically generate local results for a query). Shopping and Travel queries are likely to relate either to a user’s situational context (their primary activity at the time of querying), or to their location. Example Shopping queries include “Rapids water park coupons” which may indicate the user is about to enter a water park, and “black Converse shoes” which may indicate she would like to compare shoe prices. Queries such as “Costco” and “Walmart” also fall in the Shopping category, but likely relate to a user’s location, as the My Location feature automatically generates local results for these queries. Likewise, Travel queries such as “Metro North train schedule” and “flight tracker” may relate to a user’s situational context, and queries such as “Las Vegas tourism” may relate to their location.

In summary, an examination of the category distribution by input method of the query shows the following salient differences:

- **Voice searches are more likely to be about an “on-the-go” topic:** Mobile queries, and voice searches in particular, have a much greater emphasis on categories such as food and drink and local businesses.
- **Voice searches are less likely to be about a potentially sensitive subject:** Categories that consist of sensitive content (adult themes, social networking, and health) are avoided by voice search users, relatively speaking. This may be because they wish to preserve their privacy in a public setting.
- **Voice searches are less likely to be for a website that requires significant interaction:** Voice searches are relatively rarely about the sorts of topics that require significant interaction following the search, such as games and social networking.

4.5.1.2 By Other Attributes

We built straightforward classifiers to detect whether a query contains a geographical location, whether a query is a natural language question (such as “Who is the President of the United States?”), and whether the query is simply for a URL such as “amazon.com.” The results for the same sample of query data used earlier is illustrated in the following chart (Fig. 4.17):

Queries that include a location term such as “Mountain View” are more popular in voice search than in typed mobile search, reinforcing the result above about the broad category of local services being more popular.

Question queries, defined simply as queries that begin with a “wh” question word or “how,” are far more popular in the voice context. This may reflect the tendency of the speech user to feel more like they are in a dialog than issuing a search engine query. Another explanation is questions of this sort arise more frequently in an on-the-go context (such as “settling a bet” with friends about a factual matter).

Queries for URLs such as “amazon.com” or “times.com” are rare relative to desktop search, and rarer still relative to typed mobile queries. This reflects the fact that users tend to want information directly from their voice search experience; they are far less likely to be in a position to interact with a web site.

Reinforcing this, we have found that users are less likely to click on links returned from voice search queries than they are to click on links returned from desktop search queries, even accounting for recognition errors. This appears to be primarily because users more often pose queries that can be answered on the results page itself. Also, given current mobile network speeds, users may be reluctant to wait for an additional web page to load. Finally, voice search users are more likely to be in a context in which it’s difficult to click at all.

Finally, we find that short queries, in particular 1 and 2 word queries, are relatively more frequent in voice searches than in typed searches, and longer queries (> 5 words) are far rarer. As a result, the average query length is significantly shorter for spoken queries: 2.5 words, as compared with 2.9 words for typed mobile search and 3.1 words for typed desktop search. This result may seem counterintuitive, given that longer

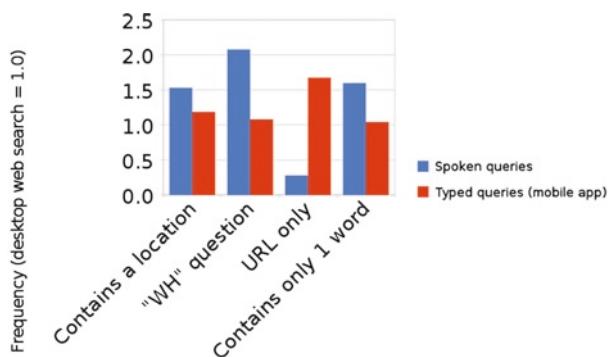


Fig. 4.17 Attribute of query

queries should be relatively easier to convey by voice. There are numerous possible explanations for this that we have not fully explored. For example, users may avoid longer queries because they are harder to “buffer” prior to speaking; or, the popular queries within the topics favored by voice search users may themselves be shorter.

4.5.1.3 Keyboard Considerations

In characterizing what distinguishes voice queries from typed queries, it matters, of course, what kind of keyboard is available to the user. For some insight on this we can look at the subset of our data from users of the BlackBerry version of GMA, which serves a number of different phone models. BlackBerry phones have two common keyboard types: a full qwerty keyboard which assigns one letter per key, and a compressed keyboard which assigns two letters for most of the keys.

Compressed keyboards make query entry more inefficient because more keypresses are needed on average to enter a query. This becomes clear when we look at the fraction of queries that are spoken rather than typed on these models, which favors the compressed keyboard by 20% relative (Table 4.6):

4.5.2 When do Users Continue to Speak?

The earlier section gave a glimpse of the characteristics of voice queries. This section looks at what factors influence whether a user chooses to search by voice rather than typing their query when they have both options available to them. Do certain factors cause them to give up using voice search entirely? Whether or not a user continues to view voice as a viable option was a matter of great importance in prioritizing efforts in the early stages of developing this product.

We have systematically measured the factors that influence whether a user continues to search by voice, and found that recognition accuracy is the most important factor. There is a strong positive relationship between recognition accuracy and the probability that a user returns, more so than other factors we considered – latency, for example – though these factors matter too.

Our analysis technique is to look at the behavior of a group of (anonymous) users in two different time intervals, spaced a short time apart, to see if certain factors were correlated with the users staying or dropping out between the two intervals. Figure 4.18 summarizes the finding for recognizer confidence, which

Table 4.6 A comparison of voice search usage for BlackBerry keyboard types

| Keyboard type | Percentage of users | Percentage of queries that are spoken |
|---------------|---------------------|---------------------------------------|
| Full | 86.9% | 34.6% |
| Compressed | 13.1% | 41.6% |

The data is based on a sample of 1.3 million queries from a 4-week period in the summer of 2009

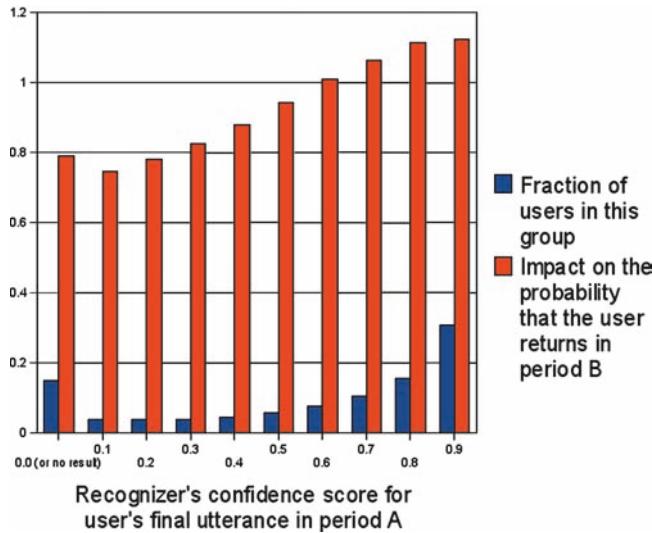


Fig. 4.18 User growth as a function of recognizer confidence

we use as a proxy for accuracy. Here, “period A” is the first week of November 2009, and “period B” is the third week of November 2009. Each of the 10 buckets along the x axis signifies the subset of users in period A whose final voice query had a particular confidence estimate according to the recognizer. The blue bar signifies the fraction of users in each bucket; for example, about 30% of users had a final query with confidence greater than 0.9. The red bar signifies the fraction of these users who returned to make voice searches in period B, normalized by our overall average retention rate. In summary, the users with final confidence greater than 0.6 were more likely than average to continue using voice search, and the other users, less likely.

4.6 Conclusions

The emergence of more advanced mobile devices, fast access to incredibly accurate (or high quality) search engines and a powerful server side infrastructure made mobile computing possible. Speech is a natural addition and provides a whole new way to search the web.

To this end we have invested heavily in advancing the state of the art. The combination of a vast amount of data resources, computational resources, and innovation has created an opportunity to make speech as common place and useful as any other input modality on the phone.

Acknowledgments Google search by Voice is a culmination of years of effort in the speech and mobile group at Google. The authors would like to thank all those involved in working on this project and making Google search by voice a reality.

References

1. C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. OpenFst: A general and efficient weighted finite-state transducer library. *Lecture Notes in Computer Science*, 4783:11, 2007.
2. M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope. Deploying GOOG-411: Early lessons in data, measurement, and testing. In *Proceedings of ICASSP*, pp 5260–5263, April 2008.
3. MJF Gales. Semi-tied full-covariance matrices for hidden Markov models. 1997.
4. B. Harb, C. Chelba, J. Dean, and G. Ghemawhat. Back-off language model compression. 2009.
5. H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.
6. M. Kamvar and S. Baluja. A large scale study of wireless search behavior: Google mobile search. In *CHI*, pp 701–709, 22–27 April 2006.
7. S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 35, pp 400–401, March 1987.
8. D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted MMI for model and feature-space discriminative training. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.
9. R. Sproat, C. Shih, W. Gale, and N. Chang. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404, 1996.
10. C. Van Heerden, J. Schalkwyk, and B. Strope. Language modeling for what-with-where on GOOG-411. 2009.

Chapter 5

“Well Adjusted”: Using Robust and Flexible Speech Recognition Capabilities in Clean to Noisy Mobile Environments

Sid-Ahmed Selouani

Abstract Speech-based interfaces increasingly penetrate environments that can benefit from hands-free and/or eyes-free operations. In this chapter, a new speech-enabled framework that aims at providing a rich interactive experience for smartphone users is presented. This framework is based on a conceptualization that divides the mapping between the speech acoustical microstructure and the spoken implicit macrostructure into two distinct levels, namely, the signal level and linguistic level. At the signal level, a front-end processing that aims at improving the performance of Distributed Speech Recognition (DSR) in noisy mobile environments is performed. At this low level, the Genetic Algorithms (GAs) are used to optimize the combination of conventional Mel-Frequency Cepstral Coefficients (MFCCs) with Line Spectral Frequencies (LSFs) and formant-like (FL) features. The linguistic level involves a dialog scheme to overcome the limitations of current human–computer interactive applications that are mostly using constrained grammars. For this purpose, conversational intelligent agents capable of learning from their past dialog experiences are used. The Carnegie Mellon PocketSphinx engine for speech recognition and the Artificial Intelligence Markup Language (AIML) for pattern matching are used throughout our experiments. The evaluation results show that the inclusion of both the GA-based front-end processing and the AIML-based conversational agents leads to a significant improvement in effectiveness and performance of an interactive spoken dialog system.

Keywords Distributed speech recognition • Mel-frequency cepstral coefficients • Line spectral frequencies • Formants • Global system for mobile • Genetic algorithms • PocketSphinx • Artificial intelligence markup language • Mobile communications

S.-A. Selouani (✉)

Professor, Information Management Department,
Chair of LARIHS (Research Lab. in Human-System Interaction),
Université de Moncton, Shippagan Campus, New Brunswick, Canada
e-mail: sid-ahmed.selouani@umcs.ca

5.1 Introduction

The robustness of speech recognition systems remains one of the main challenges facing the wide deployment of conversational interfaces in mobile communications. It has been observed that when modifying a speech recognition system whose models were trained in clean conditions to handle real world environments, its accuracy dramatically degrades [9]. Mismatches between training and test data are the roots of this drawback. In order to face this difficulty, many techniques have been developed [4]. The state-of-the-art methods can be summarized in three major approaches. The first approach consists of pre-processing the corrupted speech prior to pattern matching in an attempt to enhance the signal-to-noise ratio (SNR). It includes noise masking [3], spectral and cepstral subtraction [5], and the use of robust features. Robust feature analysis consists of using noise-resistant parameters such as auditory-based features, Mel-Frequency Cepstral Coefficients (MFCCs) [6], or techniques such as relative spectral (RASTA) methodology [10]. The second approach attempts to establish a compensation method that modifies the pattern matching itself to account for the effects of noise. Generally, the methods belonging to this approach perform Hidden Markov Models (HMMs) decomposition without modification of the speech signal [31]. The third approach is concerned with the principle to find robust recognition patterns by integrating speech with other modalities such as gesture, facial expression, eye movements, etc. [13]. Despite these efforts to address robustness, adapting to changing environments remains one of the most challenging issues of speech recognition in practical applications. It should be noted that a broad range of techniques exists for conveniently representing the speech signal in mismatched conditions [18]. Most of these techniques assume that the speech and noise are additive in the linear power domain and the noise is stationary.

Speech recognition systems will increasingly be part of critical applications in mobile communications, and have the potential to become the means by which users naturally and easily access services [11]. Therefore, making the speech-enabled interfaces more natural is one of the crucial issues for their deployment in real-life applications. Most interfaces incorporating speech interaction fall into three broad categories. The first category includes Command and Control (C&C) interfaces that rely on fixed task-dependent grammar to provide user interaction [23]. Their main advantage is their ease of implementation and high command recognition rate. However, their downside is the high cognitive load they induce when users interact with the system because of its lack of flexibility and lack of uniform command sets. The Universal Speech Interface project tries to fix some of the problems tied to C&C and natural language processing. This is done by providing a general task independent vocabulary for interaction [21]. In this category, the user is still limited in his choice of utterance since the system is strict on form. The second category is based on interactive voice response (IVR) that guides users by the means of prompts in order to validate the utterance at every step [27]. This style of interaction is mostly used in menu navigation such as that found with phone and cable companies.

Its relative lack of efficiency for fast interaction makes it a poor choice for everyday use. Finally, the third category uses natural language processing to parse the user’s utterance and to determine the goal of the request. This can be done through multiple ways such as semantic and language processing and filtering [28]. Hence, to be effective, due to “limitless” vocabulary, this type of interface needs an accurate recognizer. Another disadvantage of this system is the relatively steep development cost. This is mainly due to the complexity of parsing spontaneous utterances that might not follow conventional grammar.

In this chapter, we present an effective framework for interactive spoken dialog systems in mobile communications based on a conceptualization that divides the mapping between speech acoustical microstructure and spoken implicit macro-structure into two distinct levels: the signal level and linguistic level. The signal level is based on a multi-stream paradigm using a multivariable acoustic analysis. The Line Spectral Frequencies (LSFs), and formant-like (FL) features are combined with conventional MFCCs to improve the performance of Distributed Speech Recognition (DSR) systems in severely degraded environments. An evolutionary-based approach is used to optimize the combination of acoustic streams. The second level performs a simple and flexible pattern matching processing to learn new utterance patterns tied to the current context of use and to the user profile and preferences. For this purpose, the Artificial Intelligence Markup Language (AIML) developed by [35] [2] to create chat bots is used.

5.2 Improvement at Signal Level: The DSR Noise-Robust Front-End

Transmitting the speech over mobile channels degrades the performance of speech recognizers because of the low bit rate speech coding and channel transmission errors. A solution to this problem is the DSR concept initiated by the European Telecommunications Standard Institute (ETSI) through the AURORA project [7]. The speech recognition process is distributed between the terminal and the network. As shown in Fig. 5.1, the process of extracting features from speech signals, also called the front-end process, of a DSR system is implemented on the terminal, and the extracted features are transmitted over a data channel to a remote back-end recognizer where the remaining parts of the recognition process take place. In this way, the transmission channel does not affect the recognition system performance. The Aurora project provided a normalization of the speech recognition front-end. In the context of worldwide normalization, a consortium was created to constitute the 3G Partnership Project (3GPP). This consortium recommended the use of the *XAFE: eXtended Audio Front-End* as coder-decoder (codec) for the vocal commands. The ETSI codec is mainly based on MFCCs.

Extraction of reliable parameters remains one of the most important issues in automatic speech recognition. This parameterization process serves to maintain the

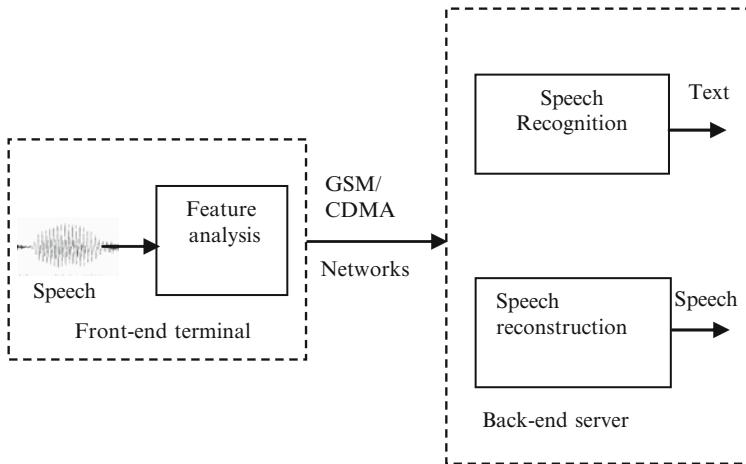


Fig. 5.1 Simple block diagram of a DSR system

relevant part of the information within a speech signal, while eliminating the irrelevant part for the speech recognition process. A wide range of possibilities exists for parametrically representing the speech signal. The cepstrum is one popular choice, but it is not the only one. When the speech spectrum is modeled by an all-pole spectrum, many other parametric representations are possible, such as the set of p -coefficients α_i obtained using Linear Predictive Coding (LPC) analysis and the set of LSFs. The latter possesses properties similar to those of the formant frequencies and bandwidths, based upon the LPC inverse filter. Another important transformation of the predictor coefficients is the set of partial correlation coefficients or reflection coefficients. In a previous work [32], we introduced a multi-stream paradigm for ASR in which we merged different sources of information about the speech signal that could be lost when using only the MFCCs to recognize uttered speech. Experiments in [33] showed that the use of some auditory-based features and formant cues via a multi-stream paradigm leads to an improvement of the recognition performance. This proved that the MFCCs lose some information relevant to the recognition process despite the popularity of such coefficients in all current speech recognition systems. In these experiments, a 3-stream feature vector is used. The first stream vector consists of the *classical* MFCCs and their first derivatives, whereas the second stream vector consists of acoustic cues derived from hearing phenomena studies. Finally, the magnitudes of the main resonances of the spectrum of the speech signal were used as the elements of the third stream vector.

In [1], we investigated the potential of the multi-stream front-end using LSFs, MFCCs, and formants to improve the robustness of a DSR system. In DSR systems, the feature extraction process takes place on a mobile set with limited processing power. On the contrary, there is a certain amount of bandwidth available for each user for sending data. Formant-like features and LSFs are more suitable for this application, because extracting them can be done as part of the process of extracting

MFCC, which saves a lot of computational process. However, due to some problems related to their inability to provide information about all parts of speech such as silence and weak fricatives, formants have not been widely adopted. In [8], it has been shown that shortcomings of formant representation can be compensated to some extent by combining them with features containing signal level and general spectrum information, such as cepstrum features.

5.2.1 *Line Spectral Frequency Cues*

Line spectral frequencies were introduced by [15]. They have been proven to possess a number of advantageous properties such as sequential ordering, bounded range, and facility of stability verification [29]. In addition, the frequency-domain representation of LSFs makes incorporation of human perception system properties easier. The LSFs were extracted according to the ITU-T Recommendation G.723.1, converting the LPC parameters to the LSFs [16]. In the LPC, the mean squared error between the actual speech samples and the linearly predicted ones is minimized over a finite interval, in order to provide a unique set of predictor coefficients.

LSFs are considered to be representative of the underlying phonetic knowledge of speech and are expected to be relatively robust in the particular case of ASR in noisy or band-limited environments. Two main reasons motivated our choice to consider the LSFs in noisy mobile communications. The first reason is related to the fact that LSF regions of the spectrum may stay above the noise level even in very low signal-to-noise ratios, while the lower energy regions will tend to be masked by the noise energy. The second reason is related to the fact that LSFs are widely used in conventional coding schemes. This avoids the incorporation of new parameters that may require important and costly modifications to current devices and codecs.

5.2.2 *Formant-Like Features*

The choice to include FL features in DSR is justified by the fact that the formants are considered to be representative of the underlying phonetic knowledge of speech and like LSFs, they are relatively robust in the particular case of speech recognition in noisy or band-limited environments. It is also well established that the first two or three formant frequencies are sufficient for perceptually identifying vowels [22]. Another advantage of using FL features is related to the formant ability to represent speech with very few parameters. This is particularly important for the systems with limited coding rate such as DSR systems. It is worth noting that many problems are associated with the extraction of formants from speech signals. For example, in the case of fricative or nasalized sounds, formants are not well defined. Several methods suggested in the literature provide a solution to the problem of determining formant frequencies. However, accurate determination of formants remains a challenging task.

There are basically three mechanisms for tracking formant frequencies in a given sonorant frame: computing the complex roots of a linear predictor polynomial; analysis by synthesis; and peak picking of a short-time spectral representation [26, 4]. In the LPC analysis, speech can be estimated in terms of a ratio of z polynomials, which is the transfer function of a linear filter. The poles of this transfer function include the poles of the vocal tract and those of the voice source. Solving for roots of the denominator of the transfer function gives both the formant frequencies and the poles corresponding to the voice source.

Formants can be distinguished by their recognized property of having relatively larger amplitude and narrow bandwidth. While this method by its very nature tends to be precise, it turns out to be expensive by virtue of the fact that in representing 4–5 formants, the order of the polynomial will often go beyond 10. Analysis by synthesis is a term used to refer to a method in which the speech spectrum is compared to a series of spectra that are synthesized within the analyzer. In such systems, a measure of error is computed based on the differences between the synthesized signal and the signal to be analyzed. The process of synthesizing signals continues until the smallest value of error is obtained. Then, the properties of the signal that generated the smallest error are extracted from the synthesizer. In the case of formant tracking, this information contains the formant frequencies and bandwidth. The third approach, which is more typical than the other two approaches, consists of estimating formants by the peaks in the spectral representation from short-time Fourier transform, filter bank outputs, or linear prediction. The accuracy of such peak-picking methods is approximately 60 Hz for the first and second formants and about 110 Hz for the third formant. In spite of the fact that this approach provides less accurate results when compared to the other two approaches, it is nevertheless quite simple. This can be highly beneficial for real time recognizers and those with limited processing power. In DSR, using algorithms with minimum amount of computation and with minimum delay is crucial. Hence, the peak-picking algorithm is more suitable for this purpose.

Typically, an LPC analyzer with the order of 12 was used to estimate the smoothed spectral peak and then four spectral peaks were selected using a peak-picking algorithm which merely compares each sample with the two neighboring samples. The process of extracting formant-like features based on the LPC analysis is illustrated in Fig. 5.2.

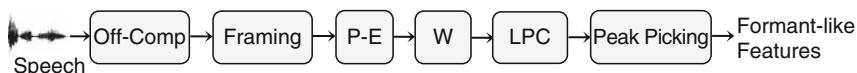


Fig. 5.2 Block diagram of a formant extractor based on the LPC analysis. The Off-comp is a block which removes the offset component, the P-E is the Pre-emphasis filter, W is the Hamming-based windowing and the LPC is the linear predictive coding analyzer providing the smoothed LPC spectrum on which the peak-peaking process is performed

5.2.3 Multi-Stream Statistical Framework

Markov Models constitute the most successful approach developed for large-vocabulary recognition systems. The statistical variations of speech are modeled by assuming that speech is generated by a Markov process with unknown parameters. A Markov process is a system that can be described at any index of time as being in one of a set of N distinct states. This system undergoes a change of state with respect to the probabilities associated with the states. In such a system, a probabilistic description requires the specification of all predecessor states as well as the current state at instant t . The HMMs that are used in speech recognition systems are first-order Markov processes in which the likelihood of being in a given state depends only on the immediately prior state. HMMs usually represent sub-word units, either context independent or context dependent, which serve to limit the amount of training data required for modeling utterances. In the multi-stream configuration, the output distribution associated with each state is dependent on several statistically independent streams. Assuming an observation sequence O composed of S input streams O_s possibly of different lengths, representing the utterance to be recognized, the probability $b_j(O)$ of the composite input vector O_t at a time t in state j can be calculated by multiplying the exponentially weighted individual stream probabilities $b_{js}(O_{st})$. Thus, $b_j(O)$ can be written as follows:

$$b_j(O_t) = \prod_{s=1}^S [b_{js}(O_{st})]^{\gamma_{js}}, \quad (5.1)$$

where O_{st} is the input observation vector in stream s at time t and γ_{js} is the stream weight. This weight specifies the contribution of each stream to the overall distribution by scaling its output distribution. The value of γ_{js} is assumed to satisfy the constraints:

$$0 \leq \gamma_{js} \leq 1 \quad \text{and} \quad \sum_{s=1}^S \gamma_{js} = 1. \quad (5.2)$$

Each individual stream probability $b_{js}(O_{st})$ is represented by the most common choice of distribution, the multivariate mixture Gaussian model, which can be determined from the following formula:

$$b_{js}(O_{st}) = \prod_{s=1}^S \left[\sum_{m=1}^M c_{jsm} N(O_{st}; \mu_{jsm}; \Sigma_{jsm}) \right]^{\gamma_{js}}, \quad (5.3)$$

where M is the number of mixture components, c_{jsm} is the m th mixture weight of state j for the source s , and N denotes a multivariate Gaussian with μ_{jsm} as the mean vector and Σ_{jsm} as the covariance matrix:

$$N(O_{st}; \mu_{jsm}, \Sigma_{jsm}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{jsm}|}} \exp^{-\frac{1}{2}(O_{st} - \mu_{jsm}) \Sigma_{jsm}^{-1} (O_{st} - \mu_{jsm})} \quad (5.4)$$

The choice of the exponents plays an important role. The performance of the system is significantly affected by the values of γ . More recently, exponent training has

received attention and the search for an efficient method is still ongoing. Most of the exponent training techniques in the literature have been developed in logarithmic domain [24]. By taking the log of the distribution function, the exponents appear as scale factors of the log terms.

$$\log b_j(O_t)^\gamma = \gamma \log b_j(O_t). \quad (5.5)$$

The multi-stream HMMs presented have three streams and therefore s is equal to three. Obtaining an estimate for the exponent's parameters is a difficult task. All HMM states are assumed to have three streams. The first two streams are assigned to MFCCs and their first derivatives, and the third stream is dedicated to LSFs or FL features. It should be noted that in order to avoid complexity, the stream exponents are generalized to all states for all models. The determination of the optimal stream weights is a critical issue. Usually, these weights are fixed empirically through cross-validation methods. In this chapter, a new approach to find the optimal stream weights by using Genetic Algorithms (GAs) is presented. In this approach, the weights are considered as individuals that evolve within an evolutionary process.

5.2.4 Evolutionary Inspired Robustness Technique

The principle of GAs consists of maintaining and manipulating a population of solutions and implementing a “survival of the fittest” strategy in their search for better solutions. The fittest individuals of any population are encouraged to reproduce and survive to the next generation, thus improving successive generations. However, a proportion of inferior individuals can, by chance, survive and also reproduce. A more complete presentation of GAs can be found in the book of [20].

For any GA, a chromosome representation is needed to describe each individual in the population. The representation scheme determines how the problem is structured in the GA and also determines the genetic operators that are used. Usually, speech applications involve genes from an alphabet of floating point numbers with values within the upper and lower bound variables. The real-valued GAs are preferred to binary GAs, since real-valued representation offers higher precision with more consistent results across replications [19].

The use of a GA requires six fundamental issues: the chromosome representation, the selection function, the genetic operators making up the reproduction function, the creation of the initial population, the termination criteria, and the evaluation function.

5.2.4.1 Initial and Final Conditions

The ideal, zero-knowledge assumption is to start with an initial population composed of the three weight sets. We choose to end the evolution process when the population reaches homogeneity in performance. In other words, when we observe that offspring do not surpass their parents, the evolution process is terminated. In our work here,

our “stop” criteria can be viewed as the convergence which accords with a stabilization of the performance, which is reflected by the phone recognition rate.

5.2.4.2 Evolving Process

In order to keep evolving strategies simple while allowing adaptation behavior, stochastic selection of individuals is used. The selection of individuals (weights) to produce successive generations is based on the assignment of a probability of selection, P_j , to each individual, j , according to its fitness value. The roulette wheel selection method can be used [12, 20]. The probability P_j is calculated as follows:

$$P_j = \frac{F_j}{\sum_{k=1}^{PopSize} F_k}, \quad (5.6)$$

where F_k equals the fitness of individual k and $PopSize$ is the population size. The fitness function will be the phoneme recognition obtained on predefined utterances (a part of test corpus) when the weight candidates γ_s are used. The general algorithm describing the evolution process is given in Fig. 5.3.

5.2.4.3 Genetic Operators

Genetic operators are used to create new solutions from the available solutions in the population. Crossovers and mutations constitute the basic types of operators. A crossover creates from two individuals (parents) two new individuals (offspring), while a mutation changes the genes of one individual to produce a new one (mutant).

A simple crossover method can be used. It generates a random number r from a uniform distribution and exchanges the genes of the parents (X and Y) on the children’s genes (X' and Y'). It can be expressed by the following equations:

```

Initialize the number of generations  $Gen_{max}$  and the boundaries of  $\gamma_s$ 
Generate for each stream, a population of 150 random individuals
For  $Gen_{max}$  generations Do
    For each set of streams Do
        Build the multi-stream noisy vectors using  $\gamma_s$ 
        Evaluate the phone recognition rate using  $\gamma_s$ 
    End for
    Select and Reproduce individuals
End For
Save the optimal weights obtained by best individuals

```

Fig. 5.3 Evolutionary optimization technique used to obtain the best stream weights

$$\begin{cases} X' = rX + (1-r)Y \\ Y' = (1-r)X + rY \end{cases} \quad (5.7)$$

The choice of this type of crossover is justified by the fact that it is simple and does not require information about the objective function. Therefore, the resultant offspring are not influenced by the performance of their parents. Mutation operators tend to make small random changes in an attempt to explore all regions of the solution space. In our experiments, the mutation consists of randomly selecting some components (genes) of a given percentage of individuals and setting them equal to random numbers generated by the uniform distribution.

The values of the genetic parameters were selected after extensive cross-validation experiments and were shown to perform well with all data. The detailed results of test experiments using GAs to optimize the DSR front-end are given in Sect. 5.4.

5.3 Improvement at Linguistic Level: The Pattern Matching-Based Dialog System

Recently, mobile devices are increasingly becoming more popular and more powerful than ever. The size and portability of mobile devices make them particularly effective for users with disabilities. Mobile devices can also be easily transported with wheelchairs. However, there are some limitations and disadvantages. For instance, the small buttons can be difficult to manipulate for people who are lacking manual dexterity. The stylus pens are often small and options for keyboard or mouse access are limited. The small screen size is also a disadvantage. Therefore, the use of speech technology may constitute a viable alternative to offset the interaction limitations of mobile devices.

Natural language interaction requires less cognitive load than interactions achieved through a set of fixed commands because the former is the most natural way used by humans to communicate. With this in mind, we propose an improvement to mobile speech-enabled platforms that allow the user to interact using natural language processing. This is accomplished by integrating an AIML through the Program# – alternatively known as AIMLBot – [34] framework. We have previously used Program# in an e-learning speech-enabled platform [25]. Program# can process over 30,000 categories in less than one second. The knowledge base consists of approximately 100 categories covering general and specialized topics of interaction. This is used to complement the fixed grammar. The AIML framework is used to design “intelligent” chat bots [2]. It is an XML compliant language. It was designed to create chat bots rapidly and efficiently. Its primary design feature is minimalism. It is essentially a pattern matching system that maps well with Case-Base Reasoning. In AIML, botmasters effectively create categories that consist of a pattern, the user input, a template, and the Bot’s answer. The AIML parser then tries to match what the user said to the most likely pattern and outputs the corresponding answer. Additionally, patterns can include wildcards that are especially useful for dialog systems.

Fig. 5.4 Example of AIML categories

```

<category>
    <pattern>GO TO TOPIC * </pattern>
    <template>
        GO_TOPIC <star/>
    </template>
</category>
<category>
    <pattern>* TO SEE TOPIC * </pattern>
    <template>
        <srai>
            GO TO TOPIC <star index="2"/>
        </srai>
    </template>
</category>

```

Moreover, the language supports recursion, which enables it to answer based on previous input. Figure 5.4 presents an example of AIML categories.

The first category represents the generic pattern to be matched and contains the ***pattern***, ***template*** and ***star*** tags. The “*” in the ***pattern*** represents a wildcard, and will match any input. In the ***template*** tag, the ***star*** tag represents the wildcard from the ***pattern*** and indicates that it will be returned in the output. For example, the input “Go to topic basics of object oriented programming” would output “GO_TOPIC basics of object oriented programming.”

The second category also makes use of the tag ***srai***. The ***srai*** redirects the input to another ***pattern***, in this case, the “GO TO TOPIC *” ***pattern***. The ***star*** tag with the *index* attribute set to “2” indicates that the second wildcard should be returned.

5.3.1 Speech-Enabled Mobile Platform

It is now becoming computationally feasible to integrate real-time continuous speech recognition in mobile applications. One such recognition engine is CMU’s PocketSphinx [14]. It is a lightweight real-time continuous speech recognition engine optimized for mobile devices. Platform speed is critical and often affects the choice of a speech recognition system. Various programming interfaces and systems have been developed around the SPHINX recognizers’ family. They are currently used by researchers in many applications such as spoken dialog systems and computer-assisted learning. In our case, as illustrated in Fig. 5.5, the SPHINX-II recognizer is used because it is faster than other SPHINX recognizers [30].

An application for navigating and searching the Web using the speech modality exclusively is implemented using the PocketSphinx recognizer. The major appeal of this application is that users can search the Web using dictation. In addition to this, if the users want to search for a word which is not in the lexicon, and if there is no practical need to add the new word to the system vocabulary, they can always revert to the spelling mode. This is also particularly useful for navigation purposes, as the



Fig. 5.5 Improved speech accessibility through mobile platform using PocketSphinx recognizer

users can spell out the URLs of web pages that they want to navigate to. The AIML module was integrated in order to improve user interaction with the system.

5.3.2 Automatic Learning Agent

To enable the system to learn new searching or navigating commands from the user, an automatic learning framework is developed. If a user said something that the system does not understand, it would ask the user if he wanted to add that as a new command. Through a feed-back system, which falls in the answer/action interface, the system would learn the new command and create the appropriate AIML entry for it.

The basis of the learning system can be represented by a finite state graph as illustrated by Fig. 5.6. State **S** is the starting state; it is when a user says a command that is not recognized by the system. It then goes to state **A**. The system repeats what the user said and asks if it got it right. If yes, it goes to state **B** and asks the user to speak the command part of what he said. Then it goes to state **C**. It asks to confirm (yes or no) if it understood the command. If yes, it goes to the final state

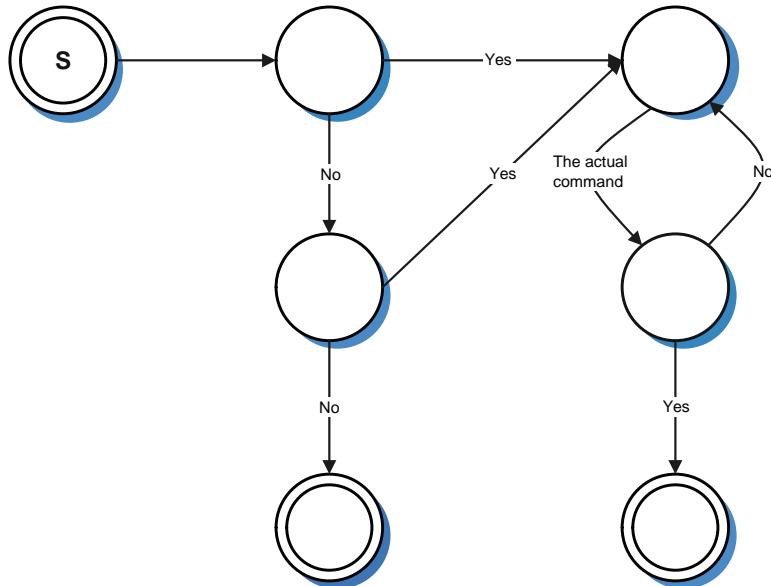


Fig. 5.6 Automatic learning finite state graph

Add which adds two entries in the AIML file with appropriate wildcards and template (dependent upon the previous context (searching or navigating)). If the user answers no in state **C**, it goes back to state **B** and proceeds until it understood the command part correctly.

There is also another possible branching in state **A**. If the user answers no to the question, it goes to state **E**. The system will then ask the user if it was a command. If yes, it goes to State **B**; if no, it goes to the final state **Quit**.

To illustrate the novelty of the proposed dialog system, the following example shows the automatic learning module in action after an unseen utterance pronounced by the user:

- User:** Could you look for chicken recipes
System: Did you say “Could you look for chicken recipes”?
User: Yes I did
System: Was it a command for me?
User: Yes
System: Could you please dictate the command part of: “Could you look for chicken recipes”
User: Could you book for
System: Did you say: “Could you book for”?
User: No
User: Could you look for
System: Did you say: “Could you look for”?
User: Yes

```

<category>
    <pattern>* could you look for *</pattern>
    <template>
        <srai>PLEASE SEARCH FOR
            <star index="2"/>
        </srai>
    </template>
</category>

<category>
    <pattern> could you look for *</pattern>
    <template>
        <srai>PLEASE SEARCH FOR
            <star/>
        </srai>
    </template>
</category>

```

Fig. 5.7 Example of created pattern after the automatic learning process

Figure 5.7 shows the creating patterns depending on the previous context that will be automatically included in the grammar set. Hence, the system learns from its experiments and errors. This behavior mimics the human learning and memorization processes.

5.4 Experiments and Results

The first goal of our experiments is to evaluate the effectiveness of GAs to provide optimal weights to the acoustic streams used in the DSR front-end. The second goal is to determine how efficient the spoken dialog system using the AIML and the automatic learning agent can be for navigating and searching the Web. The PocketSphinx recognition engine integrating the GA-optimized front-end is used for this purpose. The experimental setup is given in the following subsections.

5.4.1 AURORA Database and Baseline Systems

The AURORA database is used in the evaluation of the DSR front-end. It is a noisy speech database that was released by the Evaluations and Language resources Distribution Agency (ELDA) for the purpose of performance evaluation of DSR systems under noisy conditions. The source speech for this database is the TIDigits downsampled from 20 to 8 kHz, and consists of a connected digits task spoken by American English talkers. The AURORA training set, which is selected from the training part of the TIDigits, includes 8440 utterances from 55 male and 55 female adults that were filtered with the G.712 (GSM standard) characteristics [17].

Three test sets (A, B and C) form 55 male and 55 female adults collected from the testing part of the TIDigits from the AURORA testing set. Each set includes subsets with 1001 utterances. One noise signal is artificially added to every subset at SNRs ranging from 20 dB to -5 dB in decreasing steps of 5 dB.

Whole-word HMMs were used to model the digits. Each word model consists of 16 states with three Gaussian mixtures per state. Two silence models were also considered. One of the silence models has relatively longer duration, modeling the pauses before and after the utterances with three states and six Gaussian mixtures per state. The other one is a single-state HMM tied to the middle state of the first silence model, representing the short pauses between words. In DSR-XAFE, 14 coefficients including the log-energy coefficient and the 13 cepstral coefficients are extracted from 25 ms frames with 10 ms frame-shift intervals. However, the first cepstral coefficient and the log-energy coefficient provide similar information and, depending on the application, using one of them is sufficient. The baseline system is defined over 39-dimensional observation vectors that consist of 12 cepstral and the log-energy coefficients plus the corresponding delta and acceleration vectors. It is noted as MFCC-E-D-A. The front-end presented in the ETSI standard DSR-XAFE was used throughout our experiments to extract 12 cepstral coefficients (without the zeroth coefficient) and the logarithmic frame energy.

5.4.2 *Evaluation of the Genetically Optimized Multi-stream DSR Front-End*

To extract LSFs, 12 cepstral coefficients and the logarithmic frame energy were calculated, and then a 12-pole LPC filter and a UIT search algorithm, described in Sect. 5.2, were used. The MFCCs and their first derivatives plus the LSF vector and the log energy are referred to as MFCC12-E-D-LSF. The LSFs were combined to generate a multi-dimensional feature set. The multi-stream paradigm, through which the features are assigned to multiple streams, was used to merge the features into HMMs. In order to be consistent with the baseline, the MFCCs and their derivatives were put into the first and second streams, respectively, and the third stream was reserved for the LSFs. Tests are carried out for a configuration where the third stream is composed of 10 and 12 LSFs features. The resultant systems are, respectively, noted MFCC12-E-D-LSF10 and MFCC12-E-D-LSF12 and then, their dimensions are, respectively, 36, and 38. Equal weights are assigned to LSFs relative to MFCCs with respect to (5.14).

To extract the frequencies of FL features, a 12-pole LPC filter and a simple peak-picking algorithm were used. MFCCs and their first derivatives plus the four frequencies of the FL features were combined to generate a 30-dimensional feature set. This vector is referred to as MFCC12-E-D-4F. The multi-stream HMMs have three streams, and therefore, s is equal to three. In order to evaluate the impact of the number of included formant-like features, additional experiments where the third stream is composed of two, and three formants-like features are carried out.

Table 5.1 Values of the parameters used in the genetic algorithm

| Genetic parameters | Values |
|--|------------------|
| Number of generations | 350 |
| Number of runs | 60 |
| Population size | 150 |
| Crossover rate _(MFCCs, LSFs) | 0.30 |
| Crossover rate _(MFCCs, FLs) | 0.25 |
| Mutation rate _(MFCCs, LSFs) | 0.04 |
| Mutation rate _(MFCCs, FLs) | 0.06 |
| Boundaries of weights | [0.1, 1.5] |
| Final weights $\gamma_{s,(MFCCs, LSFs)}$ | 0.24; 0.48; 1.07 |
| Final weights $\gamma_{s,(MFCCs, FLs)}$ | 0.15; 0.66; 1.13 |

The resultant systems are respectively noted MFCC12-E-D-2F and MFCC12-E-D-3F, and then, their dimensions are respectively 28 and 29.

A set of experiments was carried out using a DSR system with the evolutionary-based optimization of the stream weights. In this configuration, multiple streams with genetically optimized weights were used to merge the features into HMMs. For instance, the frame vector that is referred to as GA-MFCC-E-D-4F is a combination of MFCCs, log energy, their first derivatives, and the four FL features. Table 5.1 presents the genetic parameters and operators used to find the optimal weights. The optimal weights are obtained after 350 generations and 60 runs. The best individual is selected. The population size is fixed at 150 individuals per generation. The optimal mutation rate is 4 and 6% for GA-MFCC12-E-D-LSF12 and GA-MFCC-E-D-4F, respectively. The best crossover rate is 30% for GA-MFCC12-E-D-LSF12 and 25% for GA-MFCC-E-D-4F. These parameters are used to obtain the final weights of each configuration.

Additional experiments were carried out in order to assess the performance of the FL features as well as the LSFs on DSR systems with gender-independent models. For each front-end, two multi-stream-based DSR systems, with the same configuration as the one using unified models for both male and female, are defined. Thus, to evaluate the performance of the LSFs features for gender-dependent speech recognition, the GA-MFCC12-E-D-LSF12F and the GA-MFCC12-E-D-LSF12M systems (referring to female and male models, respectively) are used. Similarly, to evaluate the gender-dependent DSR systems using FL features, the models for GA-MFCC12-E-D-4FF and GA-MFCC12-E-D-4FM are created. It should be noted that the same stream weights are used by both the gender-independent and the gender-dependent systems.

Table 5.2 presents the results for the babble and car noises. Best results in terms of word recognition accuracy are edited in bold. It is important to note that the multi-stream approach is more robust than the conventional DSR front-end that uses only MFCCs. For the babble noise and gender-independent models, when the SNR decreases less than 20 dB, the use of front-end composed of either LSF or FL features leads to a significant improvement in word recognition accuracy with

Table 5.2 Percentage of word accuracy of multi-stream-based DSR systems trained with clean speech and tested on set A of the AURORA database

| Signal-to-noise ratio (babble noise) | | 20 dB | 15 dB | 10 dB | 5 dB | 0 dB | -5 dB |
|--------------------------------------|---------------------------|--------------|--------------|--------------|--------------|--------------|-------------|
| MFCCs | MFCC12-E-D-A (39) | 90.15 | 73.76 | 49.43 | 26.81 | 9.28 | 1.57 |
| LSFs | MFCC12-E-D-LSF10 (36) | 85.76 | 68.68 | 44.17 | 23.55 | 11.76 | 7.41 |
| | MFCC12-E-D-LSF12 (38) | 86.46 | 68.71 | 44.89 | 23.76 | 12.09 | 7.80 |
| | GA-MFCC12-E-D-LSF12 (38) | 89.15 | 77.88 | 56.14 | 26.64 | 19.45 | 9.78 |
| | GA-MFCC12-E-D-LSF12F (38) | 88.16 | 77.56 | 55.46 | 26.24 | 19.33 | 10.2 |
| | GA-MFCC12-E-D-LSF12M (38) | 90.28 | 78.71 | 56.92 | 28.95 | 19.51 | 10.5 |
| | MFCC12-E-D-4F (30) | 87.45 | 71.74 | 52.90 | 30.14 | 12.67 | 6.92 |
| | MFCC12-E-D-3F (29) | 87.24 | 71.49 | 52.12 | 28.84 | 12.27 | 5.32 |
| FL features | MFCC12-E-D-2F (28) | 87.18 | 71.80 | 52.03 | 28.96 | 12.30 | 4.96 |
| | MFCC10-E-D-2F (24) | 75.43 | 70.14 | 51.28 | 27.92 | 11.38 | 5.16 |
| | GA-MFCC12-E-D-4F (30) | 84.52 | 76.12 | 58.74 | 35.40 | 17.17 | 9.07 |
| | GA-MFCC12-E-D-4FF (30) | 84.62 | 76.06 | 58.71 | 35.37 | 17.23 | 9.01 |
| | GA-MFCC12-E-D-4FM (30) | 80.46 | 76.56 | 58.75 | 37.40 | 19.54 | 11.5 |
| Signal-to-noise ratio (car noise) | | 20 dB | 15 dB | 10 dB | 5 dB | 0 dB | -5 dB |
| MFCCs | MFCC12-E-D-A (39) | 97.41 | 90.03 | 67.02 | 34.09 | 14.48 | 9.40 |
| LSFs | MFCC12-E-D-LSF10 (36) | 88.96 | 78.67 | 53.98 | 25.68 | 15.58 | 9.85 |
| | MFCC12-E-D-LSF12 (38) | 89.23 | 78.84 | 54.13 | 26.67 | 16.55 | 10.5 |
| | GA-MFCC12-E-D-LSF12 (38) | 91.56 | 82.65 | 68.15 | 30.12 | 18.79 | 10.7 |
| | GA-MFCC12-E-D-LSF12F (38) | 91.25 | 81.89 | 69.64 | 30.85 | 19.42 | 10.6 |
| | GA-MFCC12-E-D-LSF12M (38) | 92.74 | 83.43 | 71.06 | 32.17 | 20.36 | 10.9 |
| | MFCC12-E-D-4F (30) | 90.07 | 79.81 | 60.69 | 36.44 | 17.21 | 10.0 |
| | MFCC12-E-D-3F (29) | 89.47 | 78.91 | 59.44 | 34.63 | 17.51 | 10.3 |
| FL features | MFCC12-E-D-2F (28) | 89.59 | 79.12 | 59.26 | 34.18 | 17.33 | 10.2 |
| | MFCC10-E-D-2F (24) | 87.12 | 75.57 | 58.24 | 32.85 | 16.48 | 8.34 |
| | GA-MFCC12-E-D-4F (30) | 89.95 | 81.57 | 64.06 | 36.09 | 15.03 | 8.95 |
| | GA-MFCC12-E-D-4FF (30) | 93.93 | 87.22 | 69.75 | 40.35 | 14.89 | 8.21 |
| | GA-MFCC12-E-D-4FM (30) | 91.35 | 83.01 | 67.40 | 41.45 | 19.92 | 11.0 |

fewer parameters. The use of the proposed GA-based front-end with 30-dimensional feature vector generated from both MFCCs and FL features (GA-MFCC-E-D-4F) leads to a significant improvement in word recognition accuracy when the SNR varies from 5 dB to 15 dB. This improvement can reach 9%, relative to the word recognition accuracy obtained for the MFCC-based 39-dimensional feature vector (MFCC-E-D-A). When the SNR decreases below 5 dB, the GA-multi-stream front-end, using 12 LSF features, performs better. An improvement of more than 10% is observed for 0 dB SNR.

In order to keep the same conditions with the ETSI standard in terms of number of front-end parameters, we have carried out an experiment where we removed the two latest MFCCs and replaced them by two FL features. The DSR system remains robust. This demonstrates that for lower SNRs, we can reach better performance than the current ETSI-XAFE standard with fewer parameters. It should be noted that under high-SNR conditions, the 39-dimensional system performs better. These results suggest that it could be interesting to use concomitantly the three front-ends: the GA-LSF features under severely degraded noise conditions ($\text{SNR} \leq 0$ dB), the

GA-LF features for intermediate SNR levels ($0 \text{ dB} < \text{SNR} \leq 15 \text{ dB}$), and the current DSR-XAFE for relatively noise-free conditions ($\text{SNR} > 15 \text{ dB}$). In this case, the estimation of SNR is required in order to switch from one front-end to another.

For the car noise and gender-independent models, in the very low SNR (below 5 dB), the LSF-based DSR system is the most robust. The relatively global better results obtained in the car noise case can be explained by the complexity of the babble noise where speech interference is involved instead of pure noise. The FL features are more accurate than the LSF when the SNR is close to 5 dB. This is probably the result of the efficacy of peak prominence of formants in the noisy spectrum.

It must be noted that the genetic optimization of the stream weights yields more robustness in all contexts, including gender-dependent or gender-independent models. The comparison of gender-dependant models shows that formant representation is more efficient when the signal is severely degraded. The difference in performance of the female and male recognition multi-stream systems could be due to the limited bandwidth of the speech, which causes the expulsion of formants with frequencies greater than 4 kHz. High-frequency formants mostly appear in female speech due to shorter vocal tracts. This can result in damage to the recognition process both in training and testing. However, the gender-dependent DSR systems using the GA-based weight optimization are globally more accurate than the gender-independent systems in the context of severely degraded environments.

5.4.3 Evaluation of the Spoken Dialog System

In order to evaluate the efficacy of the spoken dialog system using the AIML-based framework, an informal dialog testing is performed. The dialog aims at navigating and searching the Web. Both classical search and navigation dialog methods and utterances are compared with the new possibilities provided by the implementation of AIML-based framework. A typical sequence of an original searching dialog is given in Fig. 5.8.

Thanks to the implementation of AIML in the robust PocketSphinx recognition system, the searching dialog can take multiple forms as illustrated in Fig. 5.9.

| |
|---|
| User: Search fo r System: Please dictate your keywords User: Banana split recipes System: Banana split recipes User: Stop System: Your keywords are banana split recipes User: Begin searching System: Searching for banana split recipe s User: Display result number 5 System: displaying result numbe r 5 |
|---|

Fig. 5.8 Typical dialogue sequence performed by current systems

| |
|---|
| <i>User:</i> Could you search for banana split recipes? |
| <i>System:</i> Searching for banana split recipe s |
| <i>User:</i> I'd like to see the fifth result |
| <i>System:</i> displaying result number 5 |
| Or: |
| <i>User:</i> Search for banana split recipes and display the top 20 results |
| <i>System:</i> Searching for banana split recipe s |
| <i>User:</i> I'd like to see result number 15 |
| <i>System:</i> displaying result number 15 |

Fig. 5.9 Possible dialogue sequences performed by the AIML-based system

As we can see from those results, the conventional system only allowed user interaction in a strictly controlled and sequential manner. However, the new system allows the user to speak more freely and naturally to the system, as instead of speaking five commands, the same result can be accomplished in only two commands. The efficacy of the new dialog approach can also be demonstrated in navigation application. To navigate to a website using a conventional dialog system, a user would have to follow this syntax:

User: Navigate
User: www.umcs.ca
User: go

The implementation of AIML-based dialog system allows multiple forms of navigation dialog:

User: I'd like to see the page at www.umcs.ca
Or:
User: go to the page at at www.umcs.ca

The AIML-based spoken dialog system allows the user to communicate in a more natural way by using intuitive utterances rather than conventional systems that use fixed commands. This leads to an improved user experience by reducing his cognition load. Indeed, the user is not required to learn any specific set of commands. The system is able to both interpret a large array of utterances and adapt to new ones by taking the current context into account.

5.5 Conclusion and Future Trends

The first mass-produced windows, icons, mouse, and pointer (WIMP)-based machine is unanimously recognized as the beginning of the popular computing. The human-computer interaction continues to play a leading role in the information and communication technology market since a product's success depends on each user's experience with it. Motivated by the expressive power of speech as a natural means of intuitive interaction, we have presented, in this chapter, a series of tools and technologies that

provide an augmented interaction modality by incorporating speech and natural language processing in mobile devices. However, incorporating speech technologies into real-life environments yields many technological challenges. The recognition systems must be sufficiently robust and flexible in order to cope with environment changes.

For this purpose, a new front-end for the ETSI DSR XAFE codec is presented. The results obtained from the experiments carried out on AURORA task 2 showed that combining cepstral coefficients with LSFs and FL features using the multi-stream paradigm optimized by genetic algorithms leads to a significant improvement of speech recognition rate in noisy environments. In the context of severely degraded environments, gender-specific models are tested and the results showed that these models yield improved accuracy over gender-independent models. In light of opportunities provided by the introduction of speech modality in mobile devices, users are faced with the task of reviewing their overall interaction strategy. Thus, from our viewpoint, universal and robust speech-enabled intelligent agents must be able to provide natural and intuitive means of spoken dialog for a wide range of applications. Moreover, the mapping between modalities and services should be dynamic. The service that is associated with the speech modality can be determined ad hoc according to the user's context, rather than being previously fixed.

The AIML-based spoken dialog system presented in this chapter proposes a flexible solution to reach this objective. The concept of command-based dialog existing in current interfaces is outdated. Instead, the information object with semantic structure will be the fundamental unit of information in future spoken dialog systems incorporated in mobile devices. User-centered interfaces will be based on more flexible information objects that can be easily accessed by their content through the use of intelligent conversational agents. Some critics say that there has been little progress in interface development since the Mac WIMP, but the author and many of his colleagues believe that a silent evolution (revolution) tends to gain a dazzling speed in order to allow intuitive and natural interaction with machines through the speech modality. Users who are given the opportunity to use dynamic and flexible speech interaction, rather than tediously typing on the small and often limited keyboards of mobile phones, should find these new tools especially attractive.

Acknowledgments This research was funded by the Natural Sciences and Engineering Research Council of Canada and the Canada Foundation for Innovation. The author would like to thank Yacine Benahmed, Kaoukeb Kifaya, and Djamel Addou for their contributions to the development of the experimental platforms.

References

1. Addou, D., Selouani, S.-A., Kifaya, K., Boudraa, M., and Boudraa, B. (2009) A noise-robust front-end for distributed speech recognition in mobile communications. *International Journal of Speech Technology*, ISSN 1381–2416, (pp. 167–173)
2. ALICE (2005) Artificial Intelligence Markup Language (AIML) Version 1.0.1, *AI Foundation*. Retrieved october 23, 2009, from <http://alicebot.org/TR/2005/WD-aiml>

3. Ben Aicha, A., and Ben Jebara, S. (2007) Perceptual Musical Noise Reduction using Critical Band Tonality Coefficients and Masking Thresholds. *INTERSPEECH Conference*, (pp. 822–825), Antwerp, Belgium
4. Benesty, J., Sondhi, MM., and Huang, Y. (2008) *Handbook of Speech Processing*. 1176 p. ISBN: 978-3-540-49128-6. Springer, New York
5. Boll, S.F. (1979) Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 29, (pp. 113–120)
6. Davis, S., and Mermelstein, P. (1980) Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4), (pp. 357–366)
7. ETSI (2003) Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithm. *Technical Report*. ETSI ES 201 (pp. 108)
8. Garner, P., and Holmes, W. (1998) On the robust incorporation of formant features into Hidden Markov Models for automatic speech recognition. *Proceedings of IEEE ICASSP*, (pp. 1–4)
9. Gong, Y. (1995) Speech recognition in noisy environments: A survey. *Speech Communications*, 16, (pp. 261–291)
10. Hermansky, H. (1990) Perceptual linear predictive (PLP) analysis of speech, *Journal of Acoustical Society America*, 87(4), (pp. 1738–1752)
11. Hirsch, H.-G., Dobler, S., Kiessling, A., and Schleifer, R. (2006) Speech recognition by a portable terminal for voice dialing. *European Patent EP1617635*
12. Houk, C.R., Joines, J.A., and Kay, M.G. (1995) A genetic algorithm for function optimization: a MATLAB implementation. *Technical report 95–09*. North Carolina University-NCSU-IE
13. Huang, J., Marcheret, E., and Visweswariah, K. (2005) Rapid Feature Space Speaker Adaptation For Multi-Stream HMM-Based Audio-Visual Speech Recognition. *Proc. International Conference on Multimedia and Expo*, Amsterdam, The Netherlands
14. Huggins-Daines, D., Kumar, M., Chan, A., Black, A., Ravishankar, M., and Rudnicky, A. (2006) Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of ICASSP*, Toulouse, France
15. Itakura, F. (1975) Line spectrum representation of linear predictive coefficients of speech signals. *Journal of the Acoustical Society of America*, 57(1), (p. s35)
16. ITU-T (1996a) Recommendation G.723.1. Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s
17. ITU-T (1996b) Recommendation G.712. Transmission performance characteristics of pulse code modulation channels
18. Loizou, P. (2007) *Speech Enhancement Theory and Practice*. 1st Edition, CRC Press
19. Man, K.F., Tang K.S., and Kwong, S. (2001) *Genetic Algorithms Concepts and Design*. Springer, New York
20. Michalewicz, Z. (1996) *Genetic Algorithms + Data Structure = Evolution Programs Adaptive*. AI series, Springer, New York
21. Nichols, J., Chau, D.H., and Myers, B.A. (2007) Demonstrating the viability of automatically generated user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1283–1292)
22. O’Shaughnessy, D. (2001) *Speech communication: human and machine*. IEEE Press, New York
23. Paek, T., and Chickering, D. (2007) Improving command and control speech recognition: Using predictive user models for language modeling. *User Modeling and User-Adapted Interaction Journal*, 17(1), (pp. 93–117)
24. Rose, R. and Momayyez, P. (2007) Integration of multiple feature sets for reducing ambiguity in automatic speech recognition. *Proceedings of IEEE-ICASSP*, (pp. 325–328)
25. Selouani, S.A., Tang-Hô, L., Benahmed, Y., and O’Shaughnessy, D. (2008) Speech-enabled tools for augmented Interaction in e-learning applications. *Special Issue of International Journal of Distance Education Technologies*, IGI publishing, 6(2), (pp. 1–20)

26. Schmid, P., and Barnard, E. (1995) Robust n-best formant tracking. *Proceedings of EUROSPEECH*, (pp. 737–740)
27. Shah, S.A.A., Ul Asar, A., and Shah, S.W. (2007) Interactive Voice Response with Pattern Recognition Based on Artificial Neural Network Approach. *International Conference on Emerging Technologies*, (pp. 249–252). IEEE
28. Sing, G.O., Wong, K.W., Fung, C.C., and Depickere, A. (2006) Towards a more natural and intelligent interface with embodied conversation agent. *Proceedings of international conference on Game research and development* (pp. 177–183), Perth, Australia
29. Soong, F., and Juang, B. (1984) Line Spectrum Pairs (LSP) and speech data compression. *Proceedings of IEEE-ICASSP*, (pp. 1–4), San Diego, USA
30. Sphinx (2009) The CMU Sphinx Group Open Source Speech Recognition Engines. Retrieved October 23, 2009 from (<http://cmusphinx.sourceforge.net/>)
31. Tian, B., Sun, M., Sclabassi, R.J., and Yi, K. (2003) A Unified Compensation Approach for Speech Recognition in Severely adverse Environment. *4th International Symposium on Uncertainty Modeling and Analysis*, (pp. 256–259)
32. Tolba, H., Selouani, S.-A., and O'Shaughnessy, D. (2002a) Comparative Experiments to Evaluate the Use of Auditory-based Acoustic Distinctive Features and Formant Cues for Automatic Speech Recognition Using a Multi-Stream Paradigm. *International Conference of Speech and Language Processing ICSLP'02*, (pp. 2113–2116)
33. Tolba, H., Selouani, S.-A., and O'Shaughnessy, D. (2002b) Auditory-based acoustic distinctive features and spectral cues for automatic speech recognition using a multi-stream paradigm. *Proceedings of the ICASSP*, (pp. 837–840), Orlando, USA
34. Tollervey, N.H. (2006) Program#- An AIML Chatterbot in C#. Retrieved August 23, 2009 from: <http://ntoll.org/article/project-an-aiml-chatterbot-in-c> Northamptonshire, United Kingdom
35. Wallace, R. (2004) The elements of AIML style. Alice AI Foundation

Part II

Call Centers

Chapter 6

“It’s the Best of All Possible Worlds”: Leveraging Multimodality to Improve Call Center Productivity

Matthew Yuschik

Matthew Yuschik, Ph d., is a Principal Investigator at MultiMedia Interfaces, LLC, and performed the research for this chapter at Convergys Corporation, Cincinnati, OH

Abstract This chapter describes trials Convergys undertook to discover how to improve call center agent productivity through the functionality provided by a multimodal workstation. The trials follow a specific sequence where multimodal building blocks are identified, investigated, and then combined into support tasks that handle call center transactions. Convergys agents tested the Multimodal User Interface (MMUI) for ease of use, and efficiency in completing caller transactions. Results show that multimodal transactions are faster to complete than only using a Graphical User Interface (GUI). Multimodal productivity enhancements are also seen to increase agent satisfaction.

Keywords Multimodality • Customer relationship management • Agent productivity • Multimodal user experience • Call center agent • GUI workstation • Human factors analysis • Call center transactions

6.1 Introduction

Convergys performed a sequence of trials designed to discover how call center agents can improve productivity by leveraging multimodal functionality on their workstation. The trials follow an experimental procedure in which multimodal building blocks are identified, investigated, and then combined to support tasks that are typical call center transactions. A larger and larger number of Convergys call center agents tested these successive refinements of a Multimodal User Interface (MMUI) to validate its ease

M. Yuschik (✉)

Senior User Experience Specialist (Multichannel Self Care Solutions),
Relationship Technology Management, Convergys Corporation, 201 East Fourth Street,
Cincinnati, Ohio 45202, USA
e-mail: yuschikholmes@comcast.net

of use and its ability to complete caller transactions. Multimodal transactions were found to be easier than using only a Graphical User Interface (GUI).

6.1.1 Convergys Call Centers

Convergys has about 85 domestic and international call centers [4]. As a global leader in relationship management, Convergys is focused on providing solutions that help clients derive more value from every customer interaction, and is continually looking for ways to improve agent productivity in the call center. One way to do this is to create a multimodal workstation for the agent which provides the opportunity to monitor agents as they process calls using multimodal capabilities. Convergys agents are efficient problem-solvers, whose experience is valuable to assess end-user needs, and to test how well a multimodal user experience can handle caller issues.

The start of this process is understanding caller behavior in customer service centers. Currently, call center agents, Subject Matter Experts in their own right, address a caller's problem by guiding the transaction through a set of screening questions to isolate the issue at hand, and then use database search tools to pose the solution to the customer. When an agent handles a customer service transaction, the constraints of the screen-based interface (the GUI) require that the agent translate the customer request into data and command terms which follow the order that the GUI expects for navigation and data entry. This requires considerable training and practice.

A multimodal interface provides additional means beyond a GUI to facilitate the call center agent's navigation and retrieval of information to complete a transaction for a caller. A Voice User Interface (VUI) adds a capability that is natural and easy to use [1, 12]. Voice and graphics (KB and mouse) can be used interchangeably to follow the existing GUI sequence of the underlying application in a step-by-step manner. This provides the flexibility to use whatever modality best matches the task at hand. Numerous human factors issues must be addressed. These issues work toward the goal that customers should be able to complete their own transaction on any device of their choice, and be highly satisfied with the result and experience.

Future predictions show that newer devices support more features and capabilities for multiple modes of interaction. These devices are richer and make for some very powerful combination of modes for specific tasks. Figure 6.1 above, illustrates that the



Fig. 6.1 Migration of multimodality from the call center to the device

testing in the call center can be leveraged to migrate to end-user mobile devices. The limiting factor is that there are no standards at this time for the devices or the network so that these multimodal phone-based capabilities can be significantly leveraged.

6.1.2 Call Centers Transactions

Convergys call centers handle about 1 billion calls per year [5]. This affords an opportunity to observe agent and caller interactions and leads to creating a model of agent behavior which includes: a *problem solving* step to isolate the key issue of the caller; *information gathering* steps with data entry and navigation through multiple GUI screens; a *resolution* step to present the solution; and a *closure* step to insure that all caller issues are addressed. Some dialog may not be directly relevant to the issue at hand, but may in fact increase caller comfort, provide optional information, or defuse caller frustration.

The call center agent is a valuable resource to evaluate any multimodal interface to complete a set of transactions. The agents understand the caller's needs and how to resolve them using existing call center methods and tools. The most important step in the process is to identify the correct flow in the call presented to the agent. The agent's actions show ways that callers can complete transactions by themselves.

Table 6.1 below shows how specific agent actions occur in typical transactions for specific call center market segments. Generic tasks and their subtasks are listed in the rows, and market segments are in the columns. The letters H, M, and L represent High, Medium, or Low frequency of occurrence of the task or subtask in the market segment.

Table 6.1 Mapping of tasks and subtasks to service

| | Health & Med. | Tech support | Shipping | Telecom | Cable & BB |
|--|---------------|--------------|----------|---------|------------|
| Navigation | | | | | |
| Specific Windows in complex CRM app | H | H | H | H | H |
| Sequential progression through workflow | L | M | H | H | M |
| Data interactions | | | | | |
| Radio button, drop-down menu, check box | M | H | H | H | H |
| Number (groups) of alphanumeric characters | M | H | H | H | H |
| Access and closure | | | | | |
| Launch support applications | L | H | M | H | H |
| Sign-on sequences | - | H | M | H | M |
| Initiation of test activities | - | VH | - | L | - |
| End, then restart application(s) | H | M | H | M | M |
| Customer record notes | | | | | |
| Transcription | VH | H | VH | H | M |
| Paste in multiple applications | M | M | L | H | L |
| Knowledgebase interactions [voice search] | | | | | |
| Retrieval of deep content | H | H | H | M | H |
| Compound query | H | M | - | M | M |

Many multimodal solutions focus on improving the efficiency or cost for a single problem, and are not structured for a larger solution. However, the overarching goal of a multimodal approach should be to create a framework that supports many solutions. Then, tasks within any specific transaction are leveraged across multiple applications. The most frequent tasks of the Table are highlighted: navigation between windows; use of radio buttons, drop-down menus, and check boxes; need to return to a start screen after every transaction; keeping words in a temporary memory; using verbal data to search retrieve significant information from storage.

6.1.3 User Versions

To appreciate the evolution path of multimodal transactions in the spectra of voice-enabled interfaces, the development of call center Interactions is described below [27].

Version 0: All calls go to live agents. This is the original call center process when a business handled every call personally. Agents used well-rehearsed scripts which they followed to consistently handle typical customer issues.

Version 1: Pressing DTMF to select options in the form “For X, Press 1,” gave callers a way to take control in their own hands to resolve simple, straightforward issues. The menu structures began from agent scripts, and choices were grouped in a tree structure for navigation using the buttons of the telephone keypad. These menus forced the caller to take part in the “solutioning” process and follow the menus. Agents still handled difficult or uncommon issues.

Version 2: “Voicify” DTMF prompts to the form “For X, Press or say 1.” This is the first foray of speech into the dialog, and is essentially a voice overlay onto Version 1. It continued the strong coupling of the transaction with the DTMF menu structure, implicitly assuming that one of the option would match the caller’s need.

Version 3: Initiate a Directed Dialog, like, “Please say listen, send or mailbox options.” This avoids the mapping from choices into numbers (to navigate menus), and lets a VUI designer present logical choices for flexible problem solving. The most common use cases are handled through voice-enabled automation, with hand off to the agent available to resolve other issues.

Version 4: Use Conversational Language to obtain a response to an open-ended question like, “What would you like to do?” The response drives the transaction by following the user’s keyword instead of imposing an automated menu structure. The interaction can drop back to *Version 3* as a way to provide options and coach to the caller to move the dialog forward. There is a risk that the caller’s issue is

not understood or cannot be resolved using the service. Once again, the agent comes to the rescue.

Version 5: Provide a multimodal approach that matches the expected transaction flow which visually displays data and options, and expects a verbal response when it asks, “What else?” Generally, voice is used for input, and text/graphics are used for output. The caller has complete control, though the system provides multiple ways for the caller to search for the resolution of their issue, starting with vague terms and then refining the search to a specific issue.

Besides adding more modalities for the agent to complete the transaction, the *User Versions* (above) show an evolution from the agent’s (inside-out) view of issues to the caller’s (outside-in) view, from a highly structured approach to an open-ended, flexible approach – flexible enough to begin with general categories, then fall back to a structure that focuses on only a few options. This strikes a balance where callers initiate the conversation with terms comfortable to them, and automation provides specific suggestions when more information or data is required to move the transaction forward. A back-and-forth dialog is maintained until all required data is obtained and the issue is resolved. This approach follows the lead of the caller, yet it is guided by automation. The interaction is dynamic (mixed initiative) versus static (menu driven).

6.1.4 Multimodal Human–Computer Interaction Styles

Multimodal interactions take on a number of styles [2, 13, 16, 18, 21]. The Human Computer Interaction (HCI) can be user-initiated, computer-initiated, mixed-initiative, or a combination of each at different places and in different manners in the transaction [10]. A user-initiated dialog involves no change to a traditional GUI-based application except to voice-enabled navigation commands and data entry for fields visible on the user’s display. A computer-initiated dialog can utilize both auditory and visual cues to focus the user’s attention on elements of the display and request specific information at each step of the transaction. Often, an MMUI initiates the dialog to get the transaction started, and then the users move the dialog forward at a rate and direction comfortable to them. This mixed-initiative approach allows the user or the computer to communicate. A variety of spoken and graphic software techniques are available to render these interactions. User style preferences are enabled with interactions that:

- actively assist the user in the most likely way to complete a task (computer-initiated action, with narrow focus), or
- suggest/show a larger set of choices for the user to select (computer-initiated, with broader focus), or
- maintain a background presence, waiting for the user to enter something (user-initiated, open focus)

Table 6.2 below shows interrelation between some telephonic devices and the modalities that are supported [24]. Devices are listed in columns, with increasing features from left to right. Modal function is shown in the rows, for both input and output communications. The mapping underscores the constraints of some devices on UIs, and the impact on how a transaction can be effectively presented.

Table 6.2 Mapping of modality capabilities to device type

| | | POTS | Cell phone | PC | PDA | 3G |
|--------|--------------|------|------------|----|-----|----|
| Input | Speak | x | x | x | x | x |
| | Type | | x | x | x | x |
| | Tap | | | x | x | |
| | GPS | | | | | x |
| Output | Listen | x | x | x | x | x |
| | Listen – TTS | | | x | x | x |
| | Read text | | x | x | x | |
| | View figures | | | x | x | x |
| | View video | | | x | | x |

6.2 Application Selection: Case Study

Bringing a multimodal service to market involves numerous business and technical steps [23]. A case study for a multimodal service, which illustrates the standard product development stages, is now discussed. The first step is identifying a likely call center service amenable to multimodality and then developing a preliminary needs assessment of the existing GUI application to determine the applicability of a multimodal interface. The second step is selecting one specific service because numerous tasks and subtasks could be implemented with multimodality, and with learning being transferred to other services with similar or identical tasks. A Wizard of Oz (WoZ) experiment was conducted as the third step which reinforced the choice of features to voice enable in the application. This led to the development of a preliminary business case, the fourth step. The WoZ implementation was improved and then placed in an environment where a limited number of agents tested it, including taking live calls. Positive results from this activity led to yet more improvements, and the fifth step, a deployment trial with a larger group of agents was compared to a control group using the existing GUI to handle callers.

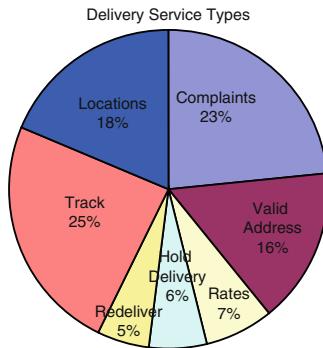
6.2.1 Matching Multimodality to an Application

A multimodal interface can voice enable all features of a GUI. This is a technologically robust solution, but does not necessarily take into account the caller's goal. Voice activating all parts of the underlying GUI of the application enables the agent

to solve every problem by following the step-by-step sequence imposed by the GUI screens. A more efficient approach, however, is to follow the way agents and callers carry on their dialog to reach the desired goal. This scenario-based (use-case) flow – with voice-activated tasks and subtasks – provides a streamlined approach in which an agent follows the caller-initiated dialog, using the MMUI to enter data and control the existing GUI in any possible sequence of steps. This goal-focused view enables callers to complete their transactions as fast as possible.

An assessment of the GUI-based interface determines if there is sufficient potential for the application to be multimodal enabled [23, 26]. The first step is to observe agents using GUI screens to complete the transactions handled in the call center. Consider a typical call center application with some voice search capabilities. A delivery service comes to mind, which accesses customer information (valid address), logistical information (package status, store locations, rates), and delivery preferences. Figure 6.2 below shows a typical distribution of transaction types for a call center delivery service.

Fig. 6.2 Delivery service transaction types



Agent observations help generate a high-level model of the call flow for each transaction type. Generally, only a small set of tasks and subtasks are required to complete the transactions. Table 6.3 shows subtasks that illustrate the basic operations which are performed with a GUI. Keyboard and mouse actions for the redelivery transaction are decomposed into the following primitive gestures.

Table 6.3 Primitive mouse and keyboard actions

-
- 1 Click on a visible button
 - 2 Move the cursor and place it in a field
 - 3 Enter data in a selected field using the keyboard
 - 4 Scroll (to expose hidden fields)
 - 5 Move cursor to a pull-down icon, and have the menu presented
 - 6 Move cursor to select element of pull-down menu
 - 7 Select date from calendar widget
-

Transaction-specific “macros” can streamline parts of transactions that improve the efficiency of the agent. For example, a novel way for an agent to return to the main menu is to have a GUI sequence that completes these steps triggered when the agent says, “Thank you for calling.”

6.2.2 *Needs Assessment*

A crucial aspect of providing a multimodal service is to enable features that match the needs of the agent. This is best structured in terms of specific transactions. Figure 6.3 (the pie chart in Sect. 6.2.1) highlights the frequency of common transactions, while the Table of Sect. 6.2.1 identifies the subtasks in those tasks. The end-user viewpoint (namely, that of the caller) is taken into consideration by designing for the usability. This places the burden on a multimodal computer service to support and anticipate actions that are necessary to complete the transactions. In particular,

1. Cover the most frequent transaction types
2. Compress and combine screens so navigation is reduced
3. Determine transaction type early, to focus only on needed info

Generic methods are required for transparently speaking data or commands. The multimodal software includes these actions in response to speech:

- Navigation – Actions and (screens) change with button, menus, and flow through parts of applications
- Intent-Oriented Navigation – groups of actions and support applications requested using a spoken keyword
- Tasks – formalize steps as a sequence using new data and standard procedures to return data toward a goal
- Repetition – highly repetitive use cases or tasks in the transaction
- Numbers – numbers with consistent format
- Data – fill multiple field with one utterance
- Menu Selection – speak from drop-down menu options (up to 20)
- Data Caching – Session-specific memory that retains information for tasks

In the call center environment, a key measure of performance is Transaction Duration (known as Average Handling Time – AHT). An application is evaluated for multimodal enablement by determining how frequent specific transactions are performed, and how much time is saved when the transaction is voice activated. This process identifies areas of high value that leverage multimodal capabilities. The entire procedure to assess time savings is to:

1. Determine the percentage of all calls in which each transaction occurs
2. Decompose tasks and subtasks of the transaction
3. Measure task and subtask completion times for GUI and VUI
4. Compute transaction-specific time savings
5. Compute overall average time saving for all transactions

This procedure identifies overall time savings as well as the transactions, tasks, and subtasks that provide the highest payback for multimodal implementation.

Voice enablement requires human factors/user experience work to review agent and customer work flow for each transaction, and identify how the caller approaches the transaction – an outside-in approach. This is compared to how the agent must handle it – an inside-out approach. Early observation and discussions with agents led to the findings that:

- 60% of data received do not follow the standard GUI screen sequence
- 35% of calls require multiple transactions which reuse information

6.2.3 Business Case Development

Transaction duration was mentioned as a key measure of performance and its improvement, and is a means to quantify reductions in agent costs. However, other value drivers which influence the business case are improved with multimodality.

Call Containment

Multimodality offers an alternative when caller are not successful with an IVR. For example, a Top 10 telecommunications company uncovered an oversight in the authentication that produced a \$4M per year savings. A Top 10 insurance provider improved only two IVR paths led to increased containment of over 400k calls per year, with associated savings of over \$1.2M.

First Call Resolution

Follow-on agent involvement gives the entire picture of how to contain almost all calls in the self-service application. Analytics help identify what is needed to resolve the problem more accurately the first time.

Increased Self Service Adoption

A leading communications company provided a speech application for “Product Instructions” to increase self-service. Adoption rate of this application averaged over 80%, and so reduced agent calls by 8,000 per month with cost reductions of >\$400k per year.

Handle Times

Obtaining customer data to resolve the caller’s issue adds to the duration length. A focused, scripted transaction flow means less time spent navigating and entering data. A top 5 company found that correcting an authentication path problem increased success rate by 10%, and so the caller need not be authenticated by the agent.

Secondary areas of performance improvement of the user experience influenced by a new multimodal Interface are:

- *Agent Productivity and Quality.* Less effort and training is needed to bring a new agent up to speed. Using speech, quality guidelines are followed when the agent repeats the information to the customer when it is spoken into the MMUI.

- *Customer Care.* The transaction is handled quicker with less effort. Caller information is accepted anytime. Standard methods for speech handle transactions more consistently
- *Agent Satisfaction.* There is increased retention since the workload is reduced, an easy-to-use script is required and the corporation is perceived as leveraging new technology. Completing transactions is easier and less stressful

Many IVR applications are Web-based. An application need not be recoded in order to integrate voice technology. A software wrapper approach overlays voice access to the GUI information in such a way that the underlying application continues to operate as if it is supporting keyboard or mouse input.

6.2.4 Business Metrics

A preliminary cost analysis for evaluating the business value of a multimodal interface for the delivery service is addressed in Table 6.4 using the key business variables.

Table 6.4 Business variables

| | |
|---------------------|------|
| Time to market | Pass |
| NVP of investment | Pass |
| Payback time | Pass |
| EPS accretive value | Pass |
| AHT savings | Pass |

Time to Market (TTM) is the time period from the deployment decision to first site deployment. It indicates how long it takes for the service to be installed at other sites, too. The Net Present Value (NVP) of the investment is the current cost of development, including hardware, software, and licenses. Payback time is the duration to recoup all start-up costs, and begin generating a positive revenue stream. Earnings Per Share (EPS) accretive value indicates prediction of future change in stock due to the service. Average Handle Time (AHT) Savings indicates the cost savings due to expected reduction in AHT due to the use of the MMUI tool. Comparing the return on investment and the payback time, the evaluation was positive and the decision was made to move forward by providing an MMUI for call center agents handling delivery service calls.

6.3 Transaction Model

Convergys constantly strives to improve customer satisfaction and increase self-service. The most efficient way to do this is by observing incoming calls, where the various aspects of flow and pace in the dialog are clearly distinguished. Most multimodal applications focus on the user interacting with a voice and graphics interface at their own pace and area of interest. An agent (e.g. customer service representative) introduces additional constraints, especially in terms of the GUI

solution sequence, translating the customer request into terms used in the GUI screens, and following the order in which the GUI expects to receive navigation and data so as to complete the service in a timely manner.

A key underlying issue is that the agent GUI screens are intended to support all possible customer transactions, while the caller is only focused on one specific issue. Overlaying multimodal onto the existing GUI workstation in the agent environment gives agents the flexibility to follow a problem-specific flow they find best and also use their voice to navigate between computer screens and populate a graphic interface.

A Goals, Operators, Methods and Selection (GOMS) Model [3, 14] is used to analyze GUI, VUI, and MMUI transactions [17, 21]. Computer applications enable the users to achieve their goals by completing a set of tasks that require successful execution of a sequence of operations. The selection of the sequential operations is called a method, and the methods can vary from person to person. For the delivery service, the operations for completing the application goals are exactly those key subtasks that were mentioned earlier and require testing and evaluation of effectiveness when voice activated. Generally, a small, covering set of core functions is required for any solution. By concentrating on this core set of voice-activated operations, a small set of actions is the focus of tuning for consistency and ease-of-use. These operations and their realizations are tested and tracked by analytics that monitor how well they enable the completion of the tasks.

6.3.1 Tasks: Steps in Completing a Goal

The agent converses with the caller to extract sufficient information and converses with the multimodal workstation to complete specific screen-based tasks. Tasks were generically shown in Table 6.1. Following the caller-focused transaction flow uses methods which leverage an MMUI. Transaction-specific flows are developed to precisely complete the tasks needed by the agent and the caller. The expectation is that callers will eventually perform the transaction by themselves on hand-held mobile devices.

Certain tasks seem to have a preferred modality – speech is excellent for input, like data entry or navigation, while graphics is better for output, like listing answers from database searches. Caller data may be obtained at any time in the conversation and placed in a speech-enabled session-specific memory until the information is required in a particular GUI field. The flows also support for backup and error handling should the solution veer off-track and require helpful redirection or restart.

The flows are tested by the agents on their multimodal workstations. Only when the transaction succeeds for the agent using it in real-world customer solutions it is considered robust enough for deployment on a smart handset. One multimodal limitation is the software currently available on mobile devices. This is a valid concern, but the availability of more 3G phones and open API software will drive the use of mobile devices to complete more complex tasks, especially those with multimodal interfaces.

6.3.2 Subtasks: Basic Parts of Tasks

Two main visual data entry mechanisms for computer applications are the keyboard, used for entry of numbers and text (e.g. name and address), and the mouse, used for a pull-down menu option, to press a button, or to scroll (move within) the current panel. Examples of these basic subtasks are listed in Sect. 6.2.1. There are equivalent speech-activated operations. Atomic operations define a basic set of subtasks to complete all tasks, and provide a basis for time comparison of a GUI and a VUI for navigation or data entry tasks. Entire transactions are decomposed using the process of Sect. 6.2.2, and then the sequence of subtasks complete the tasks used to estimate task completion time differences for GUI versus VUI modalities. This is explicitly discussed in Sect. 6.4.

6.3.3 Streamlines

Increased efficiency and ease of use are facilitated by creating multimodal streamlines (shortcuts) through the GUI [25]. A streamline is defined to be a set of GUI subtasks that completes a task within a transaction [19]. It may complete an entire task, but generally just expedites a set of substeps. A streamline is triggered by a verbal event that launches a sequence of steps normally completed one at a time using a fixed sequence of GUI screens. The streamline makes assumptions based on the transaction call flow, and acts like a “macro” to reach the solution goal quicker.

Verbal shortcuts are created, and are spoken during the solutioning step to enable the agent to jump to screens where specific data is required. This is often called Intent-Oriented Navigation. The MMUI renders a streamline by executing commands and changing screens to move the transaction to a stationary point. Any necessary parameter values can be retrieved from session-specific memory with information spoken earlier by the caller earlier or with transaction default values. A streamline can be viewed as “jumping ahead” to the next key GUI screen. (See the Figure below.) This is often at the point where a (voice) search has just been completed, and so the transaction must be reviewed and/or assumptions modified or new data entered. The net result is that the agents are not required to perform all of the GUI-imposed actions tangential to the caller’s transaction, and so they can give their full attention to the caller and the flow.

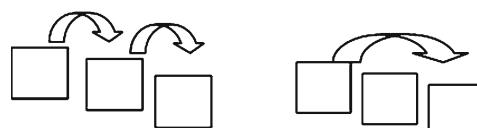


Fig. 6.3 Step-by-step compared to streamlined flows

6.3.4 *Multimodal Dialog Model*

A Multimodal Dialog Model provides visual, auditory, or combined cues to request the required input to the service. A typical GUI screen has many transaction-specific fields which can be filled. A visual cue is as simple as highlighting a required field in a particular color. An auditory cue is as simple as a whisper in the headset that asks “what service?” The GUI already supports multiple computer-initiated operations based on screen layouts, functions supported, data fields, and how they are located in terms of buttons, drop-down menus, field size, field placement, and order in which screens are presented. A GUI can direct the focus of a dialog by using background color, character color and font size, and blink rate and prepositioning of the cursor. A VUI can direct focus through an active vocabulary list and highlighted words on the GUI.

A multimodal interface for a typical call center service permits investigating the best mode for users to complete a transaction. “Best” solutions tend to follow how humans converse, keeping the caller engaged and providing relevant and necessary information. Special attention is given to determine tasks best suited to voice or to graphics to obtain the best use of a MMUI, and understanding why other modalities are less effective. Default values (terms) are best shown visually, since they provide context for review or modification. Displays are also valuable for showing new information that describes the solution more fully – an updated status, as it were. Commands are best accepted verbally, since they tend to be short and concise, with only a few words active at any one time. Command words typically trigger “state-changing” events that take the data from the current screen, execute operations, and refresh the display with updated data. These state-changing events are generally places where new data is reviewed with the caller.

One place voice input has an advantage over graphics is for data entry such as entering a number or picking a choice from a (drop-down) menu. With graphics, the cursor is moved to the field, and then the data is typed into that field. A VUI can emulate these two steps by having the user speak the name of the field and then speak the data. Even better, for some types of data, e.g. telephone numbers, the VUI does not need the field name spoken since it can infer it when a 10-digit number is spoken. Multiple grammars are active for a particular GUI screen so that any data field can be entered at any time – presenting the illusion that speech supports parallel processing!

Visual cues leverage of speech input by providing signals to indicate the GUI accepts speech for active fields. For example, the cursor is automatically placed in a specific field, and/or other fields are highlighted. The user chooses the best modality to enter data. This illustrates a multimodal, computer-initiated dialog style that expands transaction effectiveness – first, highlight a small, focused set of data choices, and then broaden the focus, highlighting more fields to suggest more speech is acceptable.

6.3.5 Error Handling

Error handling is also critical to VUI applications since ASR technology can encounter errors due to word substitution, background noise, accent variations, and other causes. VUI error handling best practices are transparent, for example, posing an auditory yes or no question (“Did you say …?” whispered) when the recognizer has low confidence. A GUI display of the top ASR choices with their confidence scores provides the speaker a means to select the intended word as well as repeat it. Words to backup to the beginning of the task, or to restart at the beginning of the transaction, are always active and provided easy error handling mechanisms using the VUI to provide safety nets for the user.

6.4 Lab Study: Subtasks

Prior to enabling an entire multimodal call center application, some lab studies were performed to validate the potential time savings. Numerous tasks and subtasks can be compared in a controlled lab environment to make predictions of agent behavior and obtain actual measurements of the time to perform the tasks [7, 9]. Results of a study comparing a GUI to a VUI for entering data into a workstation are presented [20].

6.4.1 Modality Comparisons

Comparing the type and number of underlying actions required to complete a data entry task is a good mechanism to highlight the value of multimodality. Table 6.5 below lists the perceptual, motor, and cognitive activities (viz., operations) for the subtask of hearing a number, then either entering the number through a GUI (Keyboard or Mouse) or through a VUI (spoken word). The steps were decomposed into a set of operations, each of which could be assigned a processing time. The Table shows that fewer operations are taken using speech than the keyboard/mouse.

Table 6.5 GUI vs. VUI operations for numerical entry subtask

| Step | GUI KB or mouse | Both | MMUI voice activated |
|------|----------------------------------|----------------------|----------------------------|
| 1 | | Listen to issue | |
| 2 | | Decide to input data | |
| 3 | Translate to GUI format | | Speak data |
| 4 | Find the target key | | |
| 5 | Move hand to (far/near) location | | |
| 6 | Press key or click | | |
| 7 | Repeat step 4 if necessary | | Repeat step 3 if necessary |

6.4.2 User Interface Type

This lab experiment was performed to quantify the time savings of using multimodality compared to a simple GUI. In this small laboratory test, the subject hears a number, and then sees a command on the computer display to either “Type it” or “Say it.” This models the existing case where an agent hears a number and then types it into the GUI, and it emulates the condition where the agent hears a number then speaks it into a VUI device. Test and analysis procedures are illustrative of other modality comparisons that cover typical tasks occurring when a user interacts with a computer terminal, such as:

- Saying a number versus typing a number
- Saying a button name versus clicking a button
- Entering data in a specific field by voice or by keyboard

Since both interfaces require the user to identify the problem at hand, hear appropriate data, and to decide how to input the data into the system, the cognitive load for problem solving and data-entry preparation is independent of UI.

6.4.3 Test Conditions

A VUI permits the user to directly speak the input information held in auditory short term memory. It takes less than 0.500 s to speak a 1–2 syllable word (e.g. a number). A GUI, on the other hand, requires translating (transcoding) a numeric concept (say, a spoken word, “one”) into a specific keyboard data symbol (say, “1”), finding and moving the hand to the symbol location, then pressing a key to enter the data. The cognitive translation process takes only a short time [6, 15], and locating the GUI target involves eye movement and dwelling time (if no head movement is involved). But the human sensory-motor system to produce hand movements becomes the limiting factor. The limitations on hand movement are set by perceptual and motor processes. Fitts’s Law [3] predicts how long the user takes to move the hand across the keyboard to a specific key. Then, pressing the key takes a bit more seconds. This rough analysis predicts that using a GUI will take about 1.77 s to enter one number, compared to a VUI which will take 0.5 s. The theoretical equations that define the relationship between string length and time duration for graphic and verbal input is:

$$T_g = 1.77 + 0.57(n - 1), \text{ for graphic input(GUI)}$$

$$T_v = 0.5 + 0.3(n - 1), \text{ for verbal input (VUI)}$$

These linear models predict that the GUI will have a higher initial value (y-intercept) and will have a larger change (slope) for each additional number in the string. The difference between the two methods will increase as more digits are handled.

A test of the above conditions was made in the Convergys Human Factors Lab to validate the values and the shape of the duration times, 20 number strings of various lengths were prerecorded. A small number of internal subjects ($n = 5$) who were familiar with the technology and the basis of the test performed the multimodal comparison to demonstrate the proof of concept. The presentation of numbers and string lengths was pseudorandomized and the modalities were counterbalanced so that each subject was asked to type and to say each number on the list. The completion time of the data entry subtask using a GUI versus a VUI is compared.

6.4.4 Results and Conclusions

The data from the Human Factors Lab test were decomposed into two major parts that corresponded to the response latency (the y -intercept of the line) and the time increase per additional digit (the slope). These times were easily determined from the computer log since the end time of the prompt, the beginning time of the response, and the end of the response were explicitly marked. A least-squares fit was computed on the data, and the experimental values generated the following straight line equations that described the duration for providing the user inputs:

$$T_g = 1.23 + 0.71(n - 1), \text{ for graphic input(GUI)}$$

$$T_v = 0.91 + 0.43(n - 1), \text{ for verbal input (VUI)}$$

The plots in Fig. 6.4 below compare the duration time, T , predicted by the behavioral model with the duration, T , approximated by data from the Lab study.

The graph on the left shows the duration based on the theoretical equation above for speech and keyboard, for number strings from length 1 to 7 (longer strings encounter the limits of cognitive STM [11]). The behavioral model on the left predicts a difference of 1.3 s for 1 digit and 2.9 s for 7 digits. The least-squares graph on the right shows the difference of empirical data of 0.32 s for 1 digit and 2 s for 7 digits. The intercept of the line is indicative of reaction time, and so the lab results

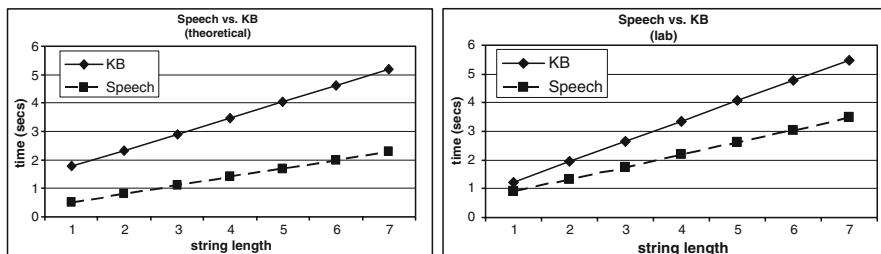


Fig. 6.4 Numerical data entered through keyboard or speech

showed a shorter time to enter the first digit with the GUI than the behavioral model while the VUI took longer time for the first digit. This would indicate the model has too much time for GUI action onset, while the VUI model may not be capturing an action. The slope is indicative of time to enter increasingly longer strings of numbers. Both GUI and VUI slopes are larger than predicted; however, the difference between the slopes is about the same for the theoretical and estimated results. The experimental results seem to match the shape and modality relationships of the theoretical model, in that it is faster to use voice input than typing numbers, and the difference in using voice gets larger as the number of digits increases.

6.5 Simulation: Tasks

Encouraged by the time savings of voice over the keyboard in a data entry subtask, a set of tasks in the delivery service were identified to validate that the results hold for larger parts of the transaction. Lateral (multiple transactions) and longitudinal (multiple trials) simulations were developed. Three different UIs were tested on these tasks, and repeated until the agent reached a steady state performance level. In addition, a small pilot test was run with an actual caller to see effects on the task durations. Lastly, a focus group was held to discuss modality preferences and areas of improvement.

Observing call center agents use a multimodal tool in a constrained learning environment is an early step in migrating an agent-enabled transaction in the call center to a self-service transaction performed by the caller on their mobile device. Analytic measurements of durations using a multimodal interface for specific delivery tasks were obtained, analyzed, and their learning effects were tracked.

6.5.1 User Interface Type

Three different UIs were investigated. The first was the currently-used GUI on which the agents had been trained. The second UI integrated voice input with existing GUI application using a “wrapper” concept, whereby software code translated spoken words into equivalent graphic counterparts required in the GUI-based transaction. The third UI used the voice capability, and included a small set of voice streamlines that freed the agent from the constraints of strictly following sequential GUI screens. The three interfaces that were tested are:

- E1 = The familiar GUI for the delivery service on which agents were trained
- AA = A voice enabled UI version that supported GUI terminology and service flow to directly substitute into the step-by-step screen-driven processes
- A2 = An expanded version of AA that included streamlined commands which followed the workflow determined from monitoring typical dialogs

6.5.2 Test Conditions

Five transactions were considered for simulation: Hold Delivery, Redeliver, Complaints, Locating a Facility, and Tracking a Package. These five delivery transactions account for about two-thirds of the call center traffic. They were broken into tasks required for their logical completion, when all information was transferred and normal closure would occur. For example, hold delivery requires starting a service request, entering a telephone number for a database search, entering a date, and then obtaining a confirmation number. While the actual workflow sequence was maintained, not all steps in a real transaction were included, specifically those requiring a caller. For example, the task of validating the caller's address as retrieved from a database was removed from the simulation. In general, there were 4 to 5 tasks associated with each simplified transaction.

The tasks were performed by 4 call center agents, 2 male and 2 female, with experience in the delivery service from 6 months to 10 years. The goal was to compare performance for the identical core call handling tasks of a transaction using different UIs. The simulated tasks were carried out on a nonproduction system without live callers to remove variability attributable to caller behavior. The agents were given caller information that was affixed to the terminal and clearly visible for the agent to enter at the appropriate points in the transaction. Removing the caller from the transaction enabled the simulation to specifically measure and compare the agent's performance.

The agents only received a small amount of training on the MMUI, which included a short description of the interface and explanation of the simplified set of tasks. A short practice session was given to insure that agent understood how to operate the voice interface properly. The agents were coached for task success since the goal was to measure the minimum time taken to complete a successful transaction. Agents were instructed to ignore any ASR errors if it did not impact transaction completion. For example, if a telephone number (TN) was misrecognized, it would be ignored if it merely retrieved different address information about the potential caller.

6.5.3 Process

The agents performed each of the five typical services, utilizing a different UI in each session on 3 successive days. The trials were videotaped. The total transaction time as well as task-specific time was measured. The order of testing the UIs (E1, AA, A1) was presented to achieve the highest possible learning transfer. E1 was an overlearned GUI, while AA was a voice-enabled version of E1; A2 relied on learning from AA as well as the use of workflow streamlines. A2 changes the emphasis from filling the fields of the GUI to that of obtaining information to complete the service. Agents used only one of the interfaces per day and performed 7 repetitions of each transaction to insure competence of using the interface was reached. Hence, 35 transactions per agent were performed in each session.

At the end of the simulation tests, a very small pilot study was performed with E1 and A2 for insight into the impact of a live caller on agent performance. A secondary goal was to observe how the agents evolved their work flow dialog to leverage the capabilities of A2. An experimenter played the role of a cooperative caller by following a script created from transcriptions of actual call center transactions. Scripts reduced the variability in results from different caller behaviors. The script contained all the caller information used in the simulation. Each of 3 agents performed 7 trials of 1 service using E1, then performed 7 trials of the same service using A2.

6.5.4 Results and Conclusions

Laboratory tests were performed in the Convergys Human Factors Lab prior to agent testing. It was found that the transaction decomposition was a solid indicator of the contributions of task durations, and that overall task duration reached its asymptote at 7 repetitions. Further, the reduced size of the active ASR grammars in the simulation led to 100% accuracy. Hence, high confidence was obtained using a small group of Convergys agents to evaluate the multimodal interface so that valuable information which would predict performance improvement of many agents was expected.

Figure 6.5a below indicates the results of the Hold Delivery simulation [18]. Very similar results were obtained for the other delivery service transactions. After a short initial transient on the simulated transaction (2–3 repetitions), agents reached the AHT asymptote, and there is very little change after that. This interface is very familiar to the agents. There is an interagent variance of about $\pm 10\%$.

Interface AA shows considerable agent variation, with a general downward trend continuing to trial 7. One agent took a longer time to accommodate the UI. The other agents showed continuing variation in the reduction of AHT. These patterns indicate that learning is still occurring after 7 trials. The quasi-steady state value of 2 agents is near E1 at trial 7. The asymptote and variance are expected to decrease slightly after more trials.

Interface A2 shows the best overall performance. There is little variance among agents, indicating their tasks are performed the same way. AHT is considerably reduced through a streamlined UI, by roughly 22%. While quasi-steady state is reached after four repetitions, a downward trend indicates that learning is still

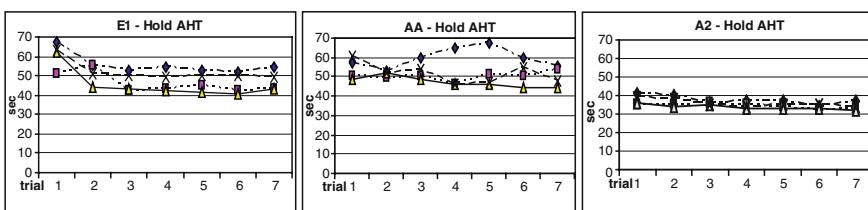


Fig. 6.5a Results for simulation of the hold transaction

taking place. Extrapolation of the trend predicts a 28% AHT reduction for the simulated tasks. While these results are highly encouraging, an overarching concern in interpreting this data is that the sample size is extremely small.

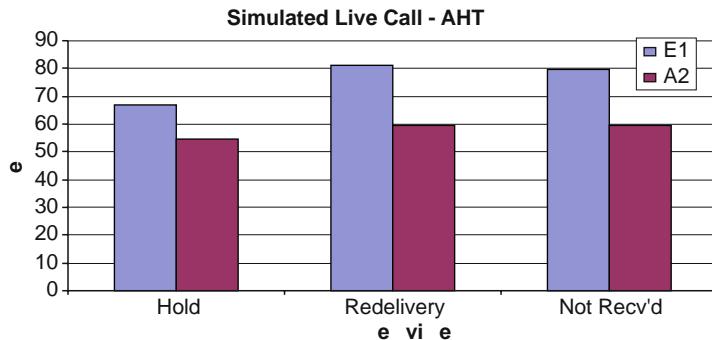


Fig. 6.5b Simulated service with caller

The results for the simulation using a dialog with a cooperative user to obtain caller information are shown in Fig. 6.5b below. Only 3 transactions were tested, with only 1 agent focusing on 1 transaction, for 7 repetitions. Since the agents were now familiar with A2, the agents concentrated on obtaining the service related data as quickly as possible. The agents explored variations of their dialogs with the caller during A2 testing, but most stabilized their dialog by the 3rd repetition. Tuning was still occurring after the 6th repetition. A very compelling result was that A2 retained all AHT savings when a caller was included into the transaction. Compared to the plots above, the data show that adding a caller increased AHT about 30 s for the E1 transaction and only about 25 s for the A2 transaction, indicating A2 worked better for work flow streamlining.

The task and subtask duration times from the simulation permit fine-tuning of the AHT values in the business case. The results improved the time-savings identified earlier, reinforced the go/no-go decision, and solidified the opinion that the multimodal interface effort should move to a larger operational trial with agents taking live calls on the call center floor.

A focus group was held the day after completion of the simulations. Agents were overwhelmingly pleased with A2, and wanted to take live calls immediately. They suggested that developers monitor live calls with agents using A2 so that any issues could be quickly resolved and new opportunities identified. The agents mentioned that they expected conversational “deadtime” when database access delay was occurring, but they felt this could be masked by talking more to the customer during these intervals, which would improve customer satisfaction. Further, increased communication would make the caller feel important, calm down an irritated caller, and be conceived as more sympathetic to caller needs. The agents did not balk at the possibility of taking more calls due to the reduction in AHT because they felt A2 made it easier to handle the calls.

6.6 Pilot: Transactions

Motivated by the success of the simulation, a pilot test was performed to validate that agents would maintain time savings when handling live calls at the call center. Using a multimodal interface while maintaining focus on the caller transaction added additional cognitive load on the agent. It is important to track the effects of a new tool on agent performance in all call center transactions so additional voice-enabled features could be added to more realistically match the agent-customer interaction.

6.6.1 User Interface Type

The MMUI is entirely consistent with the standard delivery service, from GUI screens to logical steps of service tasks; however, it also adds its own set of consistencies. A background color is used to highlight the buttons and fields that are voice-activated. Drop-down menus and radio buttons are voice-enabled. Digit strings were decomposed into nominal-sized chunks so both the speaker and the recognizer had fewer digits to process at one time. The active words at the current state are displayed on the control panel so there was no need for reference cards or job aids. When a word is spoken, the recognition result and its confidence are displayed.

Minor modifications were made to the multimodal interface evaluated in the simulation. Additional colloquial words were added to the grammars that were synonyms to the technical terms required for the GUI, and grammars were expanded to include common prolog and epilog carrier phrases in the agent's utterances (e.g. "I can help you with," "..., now"). This improved the naturalness of the agents dialog instead of being constrained to say more technical words specific to the GUI. The expanded set of words was determined by reviewing transcriptions of calls into the service and monitoring the conversations of agents during actual call handling.

A small number of MMUI streamlines – VUI “macro” commands – were added. Most were directed to expanding the types of transactions supported by the MMUI so that the agent would use the interface more ubiquitously and not feel restricted to only certain transactions. Some streamlines were tuned to accept additional relevant data of the caller, unconstrained by the sequence of the existing GUI screens. The session memory kept information spoken or retrieved during the transaction, and was also accessible by the agent's voice commands (e.g. launch a voice search using the caller's Telephone Number). Stored data could be retrieved by a streamline and populated the GUI data field when required. Streamlines were added for error handling and closure. When words or strings were repeated, they would be reentered in the GUI field (e.g. error handling for incorrect number recognition). Agents could use words like “cancel” or “backup” to restart a task from the beginning. Phrases like “How may I help you?” or “Thank you for calling” initiated new streamlines that returned to the home screen so the agent was automatically prepared to handle a new caller.

6.6.2 Trial Conditions

A group of 10 agents were selected as representatives from a team often used to test and evaluate new software releases at the call center. They were prepared to evaluate and suggest changes to the multimodal interface. They varied in age from 18 to 55, were an equal number of male and female, and had call center experience from 6 mos. to 8 years. The group included the four agents from the simulation study. The trial lasted 4 weeks, with the first two agents having also been part of the simulation. This enabled functional, network and back-end testing to validate that the continuity of these interfaces in the production environment was stable. Two more agents were phased-in after 1 week, and six more agents were added after 2 weeks.

A short, 1-h training session was provided to the agents to familiarize them with a layout and operation of the multimodal interface, describe the active vocabulary, demonstrate the action of streamlines, and answer any questions. After that, a 1-h session occurred with agents pairing-up to handle practice calls then switching roles of caller and agent. Both agents were able to view the console and ask questions of each other and the trainer. Agents then returned to their workstations on the call center floor to handle live traffic dealing with any transaction, with the MMUI operating in parallel at all time. Coaches monitored calls to the agents for the first 2–3 days, provided feedback about using the multimodal tool, showed how other transactions and tasks could be handled with multimodality, and helped remedy any ASR errors which occurred.

6.6.3 Process

The normal spectrum of live delivery service calls were routed to the agents. The agents were instructed to use the MMUI as much as possible, but balance that with their individual comfort level. AHT was the primary metric and was computed over all calls handled. It was compared to a baseline value computed from the average of daily AHT for the previous month. A Delta AHT was computed as the difference between the daily AHT in the trial and the baseline AHT. A 3-day moving average of Delta AHT was computed to smooth out the normal day-to-day variations in AHT due to varying call distribution while maintaining an accurate measure of the trend.

The “wrapper” approach was used to surround the legacy software application with a voice-enabled interface. Infrastructure variables were also monitored. Database access latencies and host delays were monitored to identify their impact on agent call handling time. ASR accuracy rate was logged and analyzed daily. Agent satisfaction was measured using a questionnaire presented at the end of the trial.

6.6.4 Results and Conclusions

Almost 17k calls handled were handled during the 21-day trial period, with the goal of obtaining over 100 calls in each transaction type so statistical analysis would have a margin of error of under $\pm 10\%$. Due to the phasing of agents onto the service and work schedule, agents had between 10 and 19 days of experience using the multimodal tool. System loading was much less than expected, with multimodal latency less than predicted (viz., negligible). ASR recognition rate was high and consistent across each agent, with a low value of 94% for one specific agent. This was caused by loud speech volume which distorted the speech signal. Additional coaching improved the accuracy for this agent to over 95%.

The performance of individual agents is shown in Fig. 6.6a below. There is no one AHT metric to characterize all agents, but the 10 agents showed different learning behavior that clustered into three distinct groups defined by their Delta AHT using multimodal technology. The number of days spent using the interface may have affected the performance. Day-to-day variations are seen, yet the daily trend of Delta AHT savings were very consistent metrics to within each group.

1. One group (Level 1) performed the best. They understood and used the interface to their advantage immediately. They showed improvement in their performance throughout the trial. They had a very short learning period, less than 7 days, after which AHT stayed below their baseline value (negative Delta AHT). There is a consistent downward trend in Delta AHT, indicating that the tool was quickly being integrated into their transaction handling.
2. Another group (Level 2) took more time (10+ days) to assimilate the tool into their work style. They learned the interface during the first 5–7 days in which their AHT increased. Their AHT then decreased below their baseline after about 12–16 days. This group took longer to reach the goal of a negative Delta AHT, but clearly showed that they were learning to use the multimodal interface, and integrating it into their call handling at a slower rate.
3. The third group (Level 3) seemed to have difficulty talking to the computer and never really took to the new multimodal interface. They generally had time using the tool, and, at best, are delayed learners of multimodality. Their initial performance was impacted negatively (Delta AHT increased at the start, and

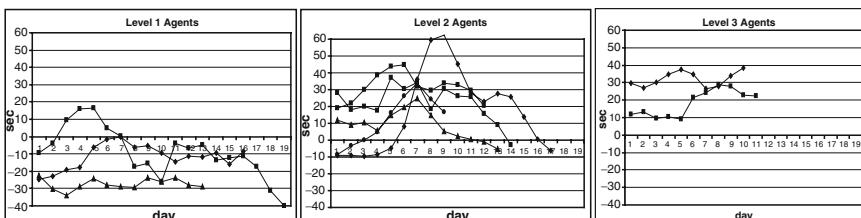


Fig. 6.6a Delta AHT for groups of agents

remained positive) and did not show other signs of improvement during the trial period. They seemed to prefer the GUI on which they were well trained.

At the end of the Pilot, 6 of the 10 agents had AHT below their baseline values, with another 2 agents trending downward. Combining the trends of Level 1 and Level 2 agents, a prediction of the performance for a larger group is possible. Figure 6.6b below repeats the Delta AHT performance for Level 2 agents. The red lines define predicted regions of time and performance. For normal learners, Delta AHT is expected to increase for about 7 days, wherein the multimodal interface is practiced enough to begin to be integrated into the agent's call handling technique. Streamlines at the beginning of the call are learned first. Then, Delta AHT decreases for the next 6 days where the multimodal interface internalized and learned to a full competence and comfort, and streamlines become a regular part of the dialog flow. Delta AHT is near zero at the end of that time. Then, the next 4 days onward are when agents utilize "deeper" multimodal capabilities and begin to reach their steady-state AHT. The target value of Delta AHT is about 10–20 s below baseline value after about 17 days of multimodal use.

Note that this prediction is for agents who are amenable to learning and using the multimodal interface. But, one size does not fit all. There is a group of agents that seem to be best suited to only using the GUI, and will never embrace multimodality.

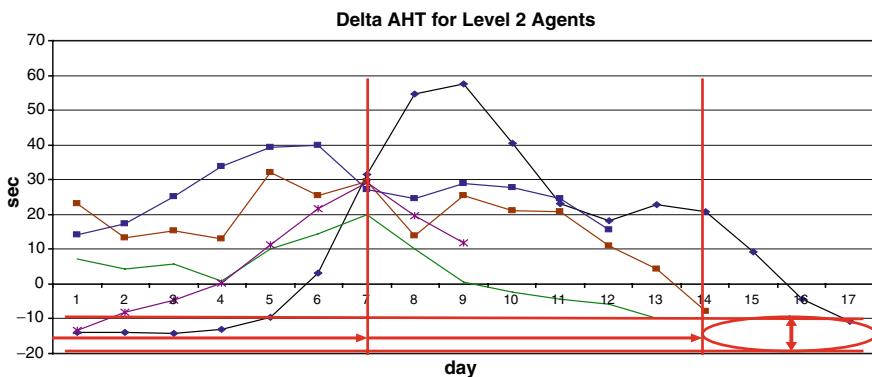


Fig. 6.6b Time and performance predictions for multimodal learning

6.6.5 Agent Satisfaction

A questionnaire was given to the call center agents that addressed the value of the multimodality as well as satisfaction in using the interface. A Likert scale was used for agents to indicate the degree to which they agreed with a statement. Agents agreed that the session memory, where intermediate data was displayed and stored, was extremely valuable in the transaction. They strongly agreed that the streamlines were valuable, were frequently used (esp., the name of the transaction), and the spoken words matched the task well. Agents strongly agreed

that there were specific actions, where the keyboard was definitely easier, and agreed that other actions were easier by voice (e.g. telephone numbers). The Team Leader of the agents commented that the starting and closure streamlines encouraged agents to follow standard customer service guidelines, which would increase customer satisfaction.

The agents were asked to suggest improvements. A need was expressed for a “mute” feature, where an agent could repeat an input to the voice system without the caller hearing it – such as, when speech was not understood, or if error handling was needed. Agents also requested more hands-on training so that the UI became more internalized, with less thought required to use it when handling live callers. A screening test was considered to determine the agents that are likely to perform well with a multimodal interface, and the topics that might require additional training.

6.7 Trial: Transactions

After retuning the formal business case, a fully-integrated functional test of multimodality was given to a larger group of agents for a longer time period. Their performance was compared to a control group that performed identical services over the duration of the trial. Both groups handled calls in the identical production environment for about 1 month. Extensive logging of system, technology, and agent performance enabled comprehensive analytics to be computed and compared.

6.7.1 *User Interface Type*

The main modifications to the prior version of the multimodal interface, including agent suggestions add more colloquial words and to support more streamlines – executing multiple actions by a single voice command. Agents also suggested introducing check-points where data must be reviewed and validated before the flow moves forward again. These changes resulted in some transactions being completed in a minimal number of steps. Almost all of the delivery service transactions were voice-enabled.

Lastly, global variables were defined to permit certain spoken utterances to be active at all times (like, “Thank you for calling,” or “How may I help you?”). These phrases returned the application to key “anchor points” when the transaction was completed or continued as part of error recovery.

6.7.2 *Conditions*

The multimodal group and the control group were matched on the demographic terms of gender, age, tenure, skill level, and AHT performance. Both groups started with 30 agents but attrition due to performance, attendance, and termination left both groups

with 27 agents. Both groups had some agents who used IVR information prior to the trial, which compromised their value as examples of typical agents. For the control group, these capabilities reduced AHT by prepopulating data received from the IVR system; for MMUI group, the capabilities were not supported, and so these agents were required to re-learn handling of particular transactions which increased their AHT. These agents in both groups were removed from the final analysis. At the completion of the month-long trial, both groups had 16 comparable agents.

A professional trainer from the Convergys staff was trained prior to the trial with three agents who used an earlier version of multimodal interface. This small group reviewed training sequence, presentations, and documents. The agents assisted in completing the functional testing of multimodality while the trainer practiced all transactions on the training system. Classroom training, for 30 agents scheduled for 3 days, was led by the professional trainer. Seven modules were covered (described in Sect. 6.8), each of which addressed and discussed a major concept of the multimodal interface, followed by practice using the concept, and then reviewing the concept through an assessment test. The written assessment tests for the first two modules were taken by each agent and evaluated by the trainer; later, modules had assessments which were completed individually; and then answers were discussed by the group as a whole. Training was performed in a “one size fits all” manner, with minor remediation done on a one-on-one basis after each individual assessment was completed. Additionally, when agents were practicing the concept, the trainer observed their performance and spent time with agent who was having difficulty.

During the 3rd day of training, agents logged on to the production system and took live calls in a controlled environment for up to $\frac{1}{2}$ h at a time. The agents were instructed to use multimodality as much as possible. Follow-up discussion of the experience was taken as a group. A 2-day transition period then followed, when agents took live calls in the classroom environment. This enabled the agents to gain multimodal experience and afforded the trainer the ability to monitor agent performance in a relatively low-key environment before agents returned to the intense activity of the call center floor. Agents could stop taking calls at any time and discuss an issue with the trainer (or an assistant coach) before returning to call handling. Observations on technique and performance-improving hints were discussed after each 1 h session.

6.7.3 *Process*

A fully-functional trial, commenced for 1 month to validate the technology infrastructure capabilities and track agent performance, shows direct and indirect benefits for multimodal features. AHT was monitored over the length of the trial. VXML Grammars supported the sets of vocabulary utterances that could be spoken at a specific context of the transaction. This helped avoid misrecognitions by reducing the number of primary choices. To further achieve better recognition, rules were included for pauses as well as “uh,” “hmm,” “um,” etc., as prologs and epilogs to a vocabulary word or phrase.

Call center data for trial and control groups was tracked by the call center ACD switch which handles all IVR calls transferred to the agents. The ACD routes calls based on agent skill from the place in the IVR where the call failed. However, data tracking at multimodal interface of the ASR server tracked the agent and each specific transaction type handled so is more accurate in identifying the service the agent provides. Many calls on the ACD and ASR server logs match, however, some are inaccurate.

Platform connectivity and infrastructure demands were also tracked. Host and server latencies for the multimodal UI and for the ASR server were monitored. ASR choices and confidence scores were logged.

Weekly feedback was provided to the agent on a standard form for speech accuracy, and AHT. The form also had information in four general areas – Home Page Commands, Numbers, Dates, and Overall Accuracy – that showed the words that were recognized correctly and that had confidence below a threshold. A comments section provided coaching hints based on the words spoken. This form provided feedback on the words spoken, how numbers were spoken, and whether dates were spoken. A plot of AHT was included, and so the agent viewed their weekly performance in terms of a familiar metric.

6.7.4 Results and Conclusions

Approximately 40,000 calls were handled by the test group during the trial, with over 1,500 calls per transaction type over the trial – yielding a $\pm 3\%$ margin of error on predictions. AHT was monitored for every agent during the days they participated in the trial, which depended on the agent's work schedule. Four key events were noticed:

- When agents moved to the call center floor, AHT increased due to the environment change. This includes taking calls without a coach to answer every question. In addition, some workstations required resetting the system and multimodal configurations.
- After the agents settled in, there is a regular weekly periodicity in call center AHT distribution. An increase in AHT occurred after a 2-day holiday, and then agents recalled how to use the interface.
- When agents were forced to always use multimodality, AHT increased and ASR recognition rate decreased. After returning to a condition, where multimodality was integrating into their workflow, many errors were eliminated and multimodality issues were avoided.
- AHT savings return after agents are told that they should use multimodality in the way that works best for them. AHT then returns to periodicity, and AHT slowly decreases. This indicates that there is a balance that works best for each agent.

Again, the agents clustered into three groups in terms of learning behavior. The metric Delta AHT (dAHT) which illustrates change from their baseline AHT is

shown in Fig. 6.7a below. This data was especially significant because it replicated earlier results in a more realistic environment.

1. One group was very successful, had immediate AHT improvement, retained the savings throughout the trial, and continued to reduce AHT. They used streamlines extensively.
2. A second group showed an initial increase in dAHT for 8–10 days when integrating the new modality into their call handling procedure. After 12 days, agents then showed negative dAHT with normal day-to-day variation and showed a continual slow trend for negative dAHT for the remainder of the trial.
3. The third group did not find the MMUI useful, and voice control seemed an intrusion into their typical call handling technique. They showed an initial increase in dAHT, and then decreased dAHT after a week on the call center floor. After that, they showed an increased dAHT. They were not getting any advantage from multimodality and seemed better suited to using their existing GUI.

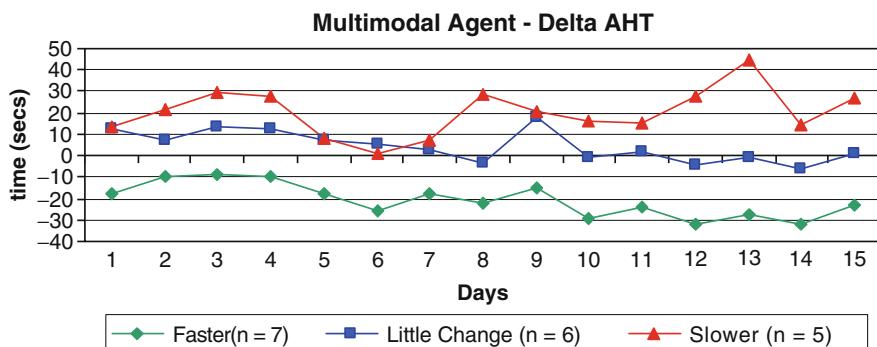


Fig. 6.7a Delta AHT for three groups of the trial. Multimodal agent delta AHT

A set of demographic features were developed as a way to note typical agent characteristics. Based on an assessment of these demographic characteristics, the three groups had specific features associated with them. The goal was to identify agents more likely to achieve transaction handling improvement with multimodality. The key characteristics of the best agents are:

- Generally younger, with shorter job tenure
- Use multimodality almost all the time
- Comfortable with many multimodal capabilities
- Speak numbers instead of typing them
- Talk to the caller and the interface simultaneously (conference mode)

Characteristics of poorly performing agents who appear to be mismatched with the multimodal interface are:

- Longer tenure
- More settled in existing procedures, resistant to change

Generally not younger agents

Write temporary information on an external notepad

Extensive use of the Mute mode when entering spoken data

Additional analytics were performed to isolate specific effects of tenure on multimodal performance. Two groups of agents were arbitrarily defined. Figure 6.7b below shows that the group of agents with less than 2 years of call center experience showed better improvement of their AHT with multimodal usage than the group with over 2 years of experience. This effect became noticeable at day 5, and there was a consistent separation between tenure groups by day 14. This difference remained throughout the trial, and AHT continued to decrease for the less tenured group.

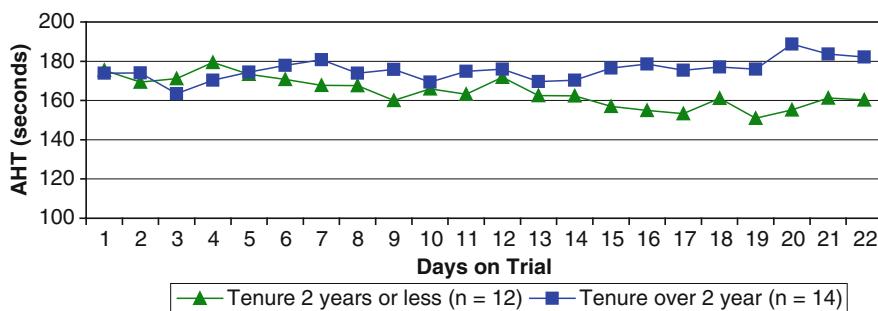


Fig. 6.7b Delta AHT for three groups of the trial. Effect of tenure on multimodal performance

Additional analytics also found that age was a factor in multimodal performance (see Fig. 6.7c below). This effect took longer to appear than that of tenure. Three age groups were arbitrarily defined. The youngest group showed consistently shorter AHT than the remainder of the agents. The AHT of this group continued to decrease throughout the trial. After day 4, there was some separation from the middle group. This is attributable to the delivery application itself having very few changes over the last 2 years, and so the older group was well trained on the current system. After day 12, there was clear separation of the youngest group from the other groups who had a harder time integrating multimodality into their call handling technique.

ASR accuracy rates were generally high throughout the trial. Overall accuracy varied between about 90 and 95%, depending on the agent and type of utterance, improving throughout the trial period to eventually reach 94+ % on the average. Accuracy includes the treatment of all speech utterances offered to the recognizer. Out-of-vocabulary speech, stuttered speech, speech restarts, and poor entry of speech (clipped begin and/or end) are pooled. Saying numbers had higher accuracy than saying navigation words, possibly due to chunking of the digits (shorter strings) by the agent – or that better agents spoke the numbers. Speech was not transcribed, and so substitution errors were not identified.

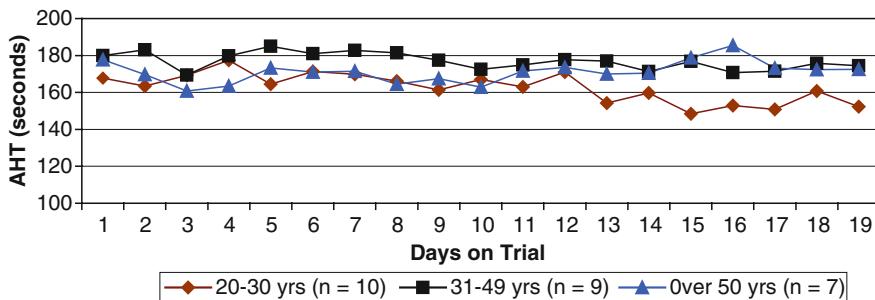


Fig. 6.7c Delta AHT for three groups of the trial. Effect of age on multimodal performance

There were a number of factors that diminished the value of comparison to the control group. In general, the control group showed a slight downward AHT trend attributable to ongoing call center AHT reduction programs. This confounded comparisons slightly between the multimodal and control groups. Further, minor changes in the call distribution of the control group occurred halfway through the trial when it was found that hold delivery and redelivery calls were only gated to selected agents of the entire group. Lastly, at the end of the trial, there were 8 multimodal agents 40 years or older compared to 5 agents in the control group so that the control group had more of the younger agents.

There were no system performance issues with the multimodal platform. The platform easily supported all the concurrent agent sessions. This was true even during training when duty cycles for ASR were much higher than when taking actual calls. While taking calls, on average, two utterances per minute were spoken per agent. There were obvious bursts of speech data (such as three chunks of a telephone number), but then longer silence periods followed while the agents reviewed retrieved information with the caller. During the trial, the multimodal platform was reconfigured to record and save all agent speech for further analysis, which led to an increase in CPU load and disk memory utilization which was well within the linear performance range of the server.

Group leaders again mentioned the improvement in agents following standard procedures for call handling. This quality measurement included saying specific terminology to the caller, repeating numbers back (for verification) and saying particular phrases to begin (“How may I help you?”) and end (“Thank you for calling.”) a call. This led to improvements in customer satisfaction. In addition, agent satisfaction increased because Convergys was viewed as a progressive company using state-of-the-art technology that intended to make the agents’ work easier.

6.7.5 Agent Preferences: Questionnaire

Subjective measures of agent satisfaction were also quantified. At the end of the trial, agents were given a questionnaire that asked for their agreement on specific

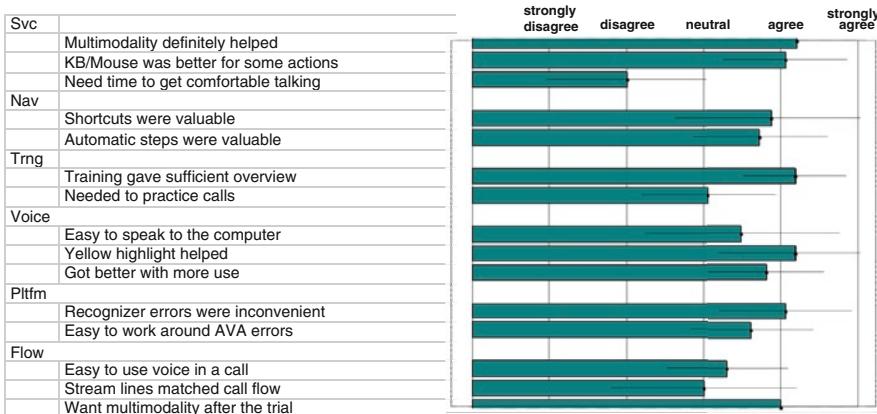


Fig. 6.8 Opinion scores of agents

aspects of the multimodal interface. There were six main categories: multimodality in general, navigation, training, voice processing, platform stability, and call flow. A Likert scale was used so the agent can indicate the degree to which he/she strongly agrees/disagrees or is neutral to the statement. The mean value and the variance of some key results are shown in Fig. 6.8 below.

The survey showed that there was clear agreement that the multimodal interface helped with call handling. There was also agreement that GUI techniques (KB and mouse) were better for some operations. Agents disagreed with the statement that it took a long time to become familiar with multimodality. There was high variance in agreement that the streamlines were valuable but smaller variance in agreement that automated steps were useful. This is likely because the high performance group strongly favored the streamlines while others used streamlines occasionally – an underlying bimodal distribution. Training was valuable. However, more time was needed to practice taking calls. While there was large variance in opinion on the ease of using speech in the interface, the agents agree that highlighting the active words on the GUI screen was valuable, as was having improved performance with increased use of multimodality. Platform errors (e.g. ASR) were viewed as inconvenient; however, agents somewhat agreed that it was easy to work around these errors. While many agents were slightly positive about ease of use and streamlines, almost all agreed that multimodality should be retained after the trial.

A number of verbatim comments made by the agents indicated high satisfaction with the trial, as is shown in Table 6.6 below. Multimodality made a big difference for hold delivery, redelivery, locations and complaints transactions, probably due to the use of streamlines. Training needed to last longer, without interruptions, and include more simulated calls for practice. Additional practice using speech input would help reduce ASR errors for short-duration utterances. And again, agents were interested in using the multimodal tool longer – many wanted the capability retained.

Table 6.6 Agent verbatims

| | |
|------------------|---|
| Service | Multimodality made big positive difference |
| Training | Need more hands-on practice with ASR and simulated calls |
| Voice processing | Sometimes necessary to repeat numbers, short one-syllable words |
| Platform | Willing to use multimodality longer |

6.8 Training

Training modules followed a model whose sequence of concepts builds on knowledge of prior modules leading to a comprehensive understanding and usage of all multimodal capabilities [22]. Each module contains numerous examples of concept description and usage, as well as exercises and/or assessment stages. Multimodal skills are structured in such a way that early training on easy-to-learn capabilities gives immediate success in reducing AHT, while subsequent training on more complex skills continues to reduce AHT further. Each step requires that a level of competence be achieved before the next skill is addressed.

6.8.1 *Training Modules*

The first module provided an overview of multimodality and its use in the delivery service. It explained that the original delivery service and all its GUI screens are still intact and can always be used. The new addition is that a speech interface (ASR) is wrapped around the GUI. The words that can be spoken (terminology), the availability of a session memory (scratchpad) that stores data until needed, and the concept of a streamline (shortcut) were introduced. A short written assessment concluded the module.

The second module addressed entering the agent's speech into the service. A push-to-talk (Walkie-Talkie) metaphor was used, in which speech is captured when a yellow button on the workstation was activated by a mouse click. A push-to-talk tool was created to practice speaking to the multimodal interface, listening to what the ASR technology heard, viewing the ASR confidence, and interpreting the utterance. A list of active words is displayed in the tool. The agent can start a transaction with a key word and then see the words available for the next step. Specific lessons address the entry of numbers. A short set of multiple choice questions assesses the agent's competence in the module.

The third module covered each multimodal screen step-by-step so the similarities between the multimodal interface and GUI interface were underscored. The first skill addressed is navigation, whether starting a streamline or returning to the home page. Then, speaking of numbers is presented, especially basic error correction. Successful error handling and returning to a success path is reviewed since it is crucial in transaction handling. Practice is undertaken and a written assessment is given.

The fourth module introduced the available streamlines which shortened the transaction by automatically performing some transaction steps. These followed

existing flows and reinforced current GUI training – use a step-by-step approach – or use a streamline to follow normal caller dialog. Stoppoints of a streamline were described as a means to review a search result with caller. Speaking longer numbers is always problematic so chunking techniques are presented, along with guidance on how to use the keypad as necessary. Numerous self-paced exercises were provided.

The fifth module introduces advanced concepts for dialog management, including a script which the agent can use to control the call flow. A checklist of best practices is provided so the agent can self-monitor performance. Then agents are paired up for role-playing where one agent practices handling a simulated call while the other evaluates the transaction using the best practices guidelines. Then, the agents reversed roles.

The sixth module discusses coaching and performance on the call center floor. This discussion is prior to taking actual calls and reminds the agent of the key steps in using multimodality to control the transaction dialog. It reinforces the value of good manners for transaction start and closure, and how these procedures improve caller satisfaction.

The seventh module is a preview of debriefing which will occur after one week of the trial. It addresses unexpected events and remaining on the success path. It also presents the debriefing process as a means to identify areas of additional training. The questionnaire of Sect. 6.7.5 is presented at that time.

6.8.2 Sequence of Modules

While the training package was effective during the short period when training-the-trainer occurred, the actual training period for the trial was interrupted repeatedly by the call center demands for agents to take live calls (using the GUI technique). With these daily interruptions, the training time was shorter than planned and adequate levels of repetition and learning was not achieved by some agents. Many key concepts were forgotten before they could be practiced and retained. Repeating the training of a module was typical. A significant result was that numerous out-of-vocabulary utterances were spoken – evidence that saying the appropriate words was not internalized.

6.8.3 Training Conclusions

The trial showed that certain agents used multimodality to its full capability right from the start while other agents did not. This may be due to training issues or risk aversion, but it is also likely that agents are learning at a different rate. Since almost all of the delivery service was voice enabled, the complexity of the resulting application may have been too high to begin with, with too much expected of the agents. However, many tasks were repeated in other transactions and repeated often. An alternative training style would be to introduce smaller multimodal parts (subtasks) to the agent, use them on live calls until comfort and competence were reached, and then move to more complex subtasks of the application.

Normal call center training assumes uninterrupted class time and that one size fits all. In reality, neither has occurred. As developed, training modules are suitable for individual sessions. The first session is best in a classroom so initial questions about multimodality are quickly resolved. Other modules are self-contained and suitable for Self-paced Computer Learning. Modules contain practice exercises and assessment tests to validate the agent's skill before continuing to the next module. This permits training to occur in the time frame that best accommodates the agent's capabilities.

6.9 Multimodal Lessons

A number of call center benefits learned from the multimodality trials include time savings in AHT, better operational compliance (quality), error mitigation, call resolution scripting, and reduced number of keystrokes. The approach also has the hidden benefit of reducing capital investment for software releases because potential GUI changes can be emulated by MMUI changes, and performance monitored to evaluate the change.

6.9.1 Best Practices

The results showed that multimodality was of most value when handling the navigation subtask, whether choosing a branch of the Home Page, initiating a search, or returning to the home page. Recognition errors were very low and easily corrected. There was little need to restrict multimodal navigation and require the GUI to execute the operation. The agents were not overwhelmed with remembering verbal choices since they already knew navigation terms from their GUI training.

Data entry into fields and selection from drop-down menus or radio buttons was also very suitable to multimodal activation. Scrolling down a text list to make a GUI selection takes considerable time for many operations. Even multimodal error handling is faster because it only requires 1 step (repeat the choice). Entering structured numerical data strings of fixed length (ZIP code, TNs) is also easy with multimodality, with very high accuracy. Error handling has at most two steps, speaking a restart term (like, "the telephone number is") and then repeating the number. Lengthy, unstructured numbers proved difficult using voice, just as it also took time with the GUI. Agents usually guide the caller to say the number in small chunks, however, callers often misread/speak long numbers and error correction is tedious. Long numbers are best suited for the GUI.

The best places to use multimodality are multistep tasks which are repeated often. One design goal is to use subtasks across multiple transactions. Another design goal is to identify transactions that can be longitudinally enabled – streamlined as it were – and completed using multimodality throughout. The key is to use sets of subtasks that are common, easy to learn, easy to use, reduce effort, and add value to the transaction.

A set of MMUI best practices is developed from the results of the delivery service. They also help provide the best techniques for other multimodal applications, such as voice search or managing financial services.

- Use a keyword to identify the transaction and start the flow of the dialog. Activate context-specific grammars for local and global content words relevant to the transaction.
- Keep a session-specific memory for information about the dialog. Store caller input as well as specific system output data. The information is retrieved when needed by the transaction.
- Set up and execute a search as soon as possible. Start with mixed initiative dialog management, and then use a directed dialog to obtain data parameters required to launch a search.
- When performing a search, give the caller feedback that relevant information is coming. For example, use a phrase like “I’ll look that up...” to set the expectation that a search is occurring.
- Present information in a logical sequence, in the order expected by the caller. The agent’s dialog should structure maximize information transfer.
- Provide “break points” where the agent can review the data, whether for caller validation or for interpreting the data, and then continue the streamline.

6.9.2 *Safety Net*

The development of an agent-based multimodal interface is the first step in the migration to software that can be given to the end-user. A small set of call center agents are given the first use of a multimodal device to handle a caller, and can always fall back to their existing workstation if there are any troublesome issues. The agents are given the task to validate success of the device in the commonly encountered use cases, and given feedback on how to improve the multimodal transaction. Usage patterns are tracked and results are analyzed to identify and address any unanticipated pain-points. Once the device-based multimodal version of a call center service handling reaches its success metrics, the software is deployed in a limited test with friendly users.

The next stage is to render the multimodal interface on end-user devices. Not all caller issues can be handled by automation, and so an effective safety net must be provided. A “Hidden agent” approach is the best monitoring and intervention mechanism (dynamic decisioning rules) that uses events to decide when the caller is having difficulty. An agent is bridged onto the call without the caller’s knowledge to move the transaction forward “behind the scenes.” The agent can view the transaction history with all the context appearing on their terminal, and/or listen to the caller’s spoken inputs. An agent’s intervention is tracked to identify areas for multimodality improvements; for situations where speech is difficult; conditions are beyond the capability of the automated solution; or when the caller’s emotional state is interfering with a solution attempt.

6.9.3 Analytics and Metrics

The multimodal user experience is influenced by factors which reflect real and perceived qualities. These qualities are maximized when following the best practices mentioned earlier (Sect. 6.9.1). Factors can be defined by variables that are measured to improve performance. Some performance metrics in Table 6.7 have already been mentioned:

Supporting these features lead to high user satisfaction based on preference surveys. However, MMUIs are not entirely understood, and so new concerns must be identified and resolved through early testing. Special attention is necessary to monitor the use of coupled modalities to insure that auditory and visual cues are complementary, and not giving “mixed signals” for data or navigation.

6.9.4 Next Steps

Convergys is actively leveraging what was learned from the agent trials and applying them in other customer-focused solutions, such as intelligent self-service solutions and development tools that support multimodal, voice and visual IVR applications. Use cases are being defined for business sectors that can take advantage of the rich multimodal environment. In the Telecom sector, call centers receive numerous calls from customers requiring help with service-impacting conditions. Whether to troubleshoot a set-top box, or listen to and download ringtones, or viewing a video or audio clip that leads to issue resolution, supporting these transactions on a mobile device has a huge advantage over an agent only being able to talk to the caller. In the retail sector, a business can display a visual rendering of the products (clothes, rental car models) and take an order using multimodal capabilities of a mobile device. The customer can select colors, sizes, etc., and view the results in a realistic environment. In the financial sector, a caller can view a pie chart of a current stock portfolio or see a listing of their banking transactions over a specific time interval. A voice search can lead to spoken or displayed information about potential investments.

Designing, prototyping, and conducting a trial of these applications provides a rich opportunity to identify those tasks and transactions suitable for migration to the ever

Table 6.7 Call center performance factors and metrics

| Factor | Description | Metric |
|------------------|--|------------------------------------|
| Control | In control of transaction at all times | Number of responses, reaction time |
| Flexibility | Various alternatives to achieve goal | Transaction success rate |
| Efficiency | Completed with minimum steps | AHT, number of steps taken |
| Self-descriptive | Colloquial terminology | Out-of-vocabulary utterances |
| Consistency | Similar actions behave identically | Subtask duration times |
| Feedback | Action have quick response | “Help” requests, repeated prompts |
| Clear exit | Cancel or backup easily accomplished | Frequency of “cancel” or “done” |

increasing number of 3G intelligent telephones. The ability of Convergys to use call center agents familiar with these transactions and willing to test alternative multimodal environments to solve caller problems is a leading-edge opportunity. Convergys is in a unique position to provide multimodal applications with cutting edge technology to match the behavioral habits of an increasingly technology-driven culture.

A streamlined view of normal transaction flow is taken as a dialog model for migrating the transaction to an end-user self-care device. The agent is brought in on-demand when users appear to have trouble. This iterative approach allows for ongoing tuning to develop the best approach to new tasks and subtasks leading to a satisfying user experience.

6.10 Future Work

Convergys is using these trial results to plan additional lab tests and other field trials. The corporation shows thought leadership in analytics, metrics, user experience, and training.

6.10.1 *Analytics and Performance*

An explicit Performance Index (PI) function must be defined to quantify the effects of numerous key variables on overall multimodal value. The major areas and variables that affect performance include the following that have been identified earlier:

- ASR accuracy – confidence value and number of reprompts
- User Satisfaction – preference score of CSAT indicators
- Transaction Completion Rate – objective task-level measurements
- Transaction Completion Time – duration of transaction for agent groups
- Modality Thrashing – change of modality for tasks or error handling
- Help Requests – location and type, repetitions

These general categories may be decomposed further into subtask variables. The PI variables are combined as linear sum of weighted variables, with initial weights of equal value. A Principle Factors Analysis will decouple dependencies and determine the most important variables, and iteratively modify the weights for an optimal representation.

6.10.2 *User Experience*

Voice modality supports the indirect extraction of required information from a dialog, enabling navigation through screens and accepting specific data entry fields as soon as possible. The agent and caller interact at their own pace in areas of

interest, which adds speed and naturalness to the completion of the transaction. Additional user experience issues are identified and tested by addressing specific use case. This approach extends lab tests of the end-user hand-held devices, where the type and the amount of information can be controlled, the type of multimodal modules can be designed and transaction complexity simplified.

Considerable value is obtained from a multimodal experience using the capabilities of a voice search. More trials are being planned using other technologies, for example, including speaker verification as a validation mechanism, and decisioning rules that adapt the dialog by applying business practices to the customer record. Techniques are implemented that highly leverage the voice mode.

6.10.3 Training

Certain agents immediately used multimodality to its full capability while others did not. This may be due to training or risk aversion. Agents may be learning at a different rate. Multimodal training is structured so easy-to-learn skills give immediate reduction in AHT, and additional training likewise continues to reduce AHT. The training process brought agents up to speed quickly and comfortably. However, it assumed three uninterrupted days of training. The content and sequence of training modules are designed to be suitable for individual lessons. The first session is best held in a classroom environment to resolve questions about multimodality quickly and provide ample encouragement about the advantages of multimodality. Other modules are self-contained and suitable for Self-Paced Computer Learning. Multimodal skills build on each other and lead to the use of all the multimodal capabilities discussed earlier. All modules contain practice exercises, with an assessment of proficiency required before the agent can continue to the next module. Training is offered with enough time to best accommodate the agent's skill set and the demands of the call center.

Acknowledgement This work was performed as part of ongoing research and development at the Convergys Corporation. Special thanks go to Jay Naik, Ph.D., Karthik Narayanaswami, Cordell Coy, Ajay Warrier, and the agents at the Convergys Call Centers.

References

1. Ballantine, B. (1999) How to Build a Speech Recognition Application, Enterprise Integration Group, San Ramon, CA
2. Brems, D., Rabin, M., and Waggett, J. (1995) Using Natural Language Conventions in the User Interface Design of Automatic Speech Recognition Systems, *Human Factors* 37(2):265–282
3. Card, S., Moran, T., and Newell, A. (1983) The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, New Jersey
4. Convergys Home Page (2010) www.convergys.com/company
5. Convergys Corporate Report (2008) www.convergys.com/investor/annual_report_2008, page 10

6. Esgate, A. and Groome, D. (2005) An Introduction to Applied Cognitive Psychology, Psychology Press, New York
7. Hauptman, A., and Rudnicky, A. (1990) A Comparison of Speech and Typed Input, CHI Minneapolis, MN
8. Heins, R., Franzke, M., Durian, M., Bayya, A. (1997) Turn-Taking as a Design Principle for Barge-In in Spoken Language Systems, *International Journal of Speech Technology* 2:155–164
9. Karl, P. and Schneiderman, B. (1993) Speech-Activated versus Mouse-Activated Commands for Word Processing Applications: An Empirical Evaluation, *International Journal of Man-Machine Studies* 39:667–687
10. Margulies, E. (2005) Adventures in Turn-Taking, Notes on Success and Failure in Turn Cue Coupling, *AVIOS2005 Proceedings*, San Francisco, CA
11. Miller, G. (1958) The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Information Processing, *Psychological Review* 63(2):81–97
12. Nass, C., and Brave, S. (2005) Wired for Speech, How Voice Activates and Advances the Human Computer Relationship, MIT Press, Cambridge, MA
13. Sacks, H., Schegloff, E.A., Jefferson, G. (1974) A Simplest Systematics for the Organization of Turn-Taking for Conversation, *Language* 50:696–735
14. Simon, H. (1978) Cognitive Psychology Class Notes, Carnegie-Mellon University, Pittsburgh, PA
15. Smith, E. and Kosslyn, S., (2007) Cognitive Psychology: Mind and Brain, Pearson Prentice Hall, Upper Saddle River, NJ
16. Yuschik, M. (1999) Design and WOZ Testing of a Voice Activated Service, *AVIOS99 Proceedings*, San Jose, CA
17. Yuschik, M. (2002) Usability Testing of Voice Controlled Messaging, *International Journal of Speech Technology* 5(4):331–341
18. Yuschik, M. (2003) *Language Oriented User Interfaces for Voice Activated Services*, Patent Number 6,526,382 B1
19. Yuschik, M., et al. (2006a) *Method and System for Supporting Graphical User Interfaces*, Patent Office Serial Number 60/882,906
20. Yuschik, M. (2006b) Comparing User Performance in a Multimodal Environment, AVIOS San Francisco, CA
21. Yuschik, M. (2007a) Silence Durations and Locations in Dialog Management, Chapter 7, Human Factors and Voice Interactive Systems, Second Edition, Bonneau and Blanchard (Eds.), Springer, New York/Heidelberg
22. Yuschik, M. (2007b) *Method and System for Training Users to Utilize Multimodal User Interfaces*, Patent Office Serial Number 60/991,242
23. Yuschik, M. (2008a) Case Study: A Multimodal Tool for Call Center Agents, *SpeechTEK2008*, D101- Design Methods and Tools
24. Yuschik, M. (2008b) Steps to Determine Multimodal Mobile Interactions, *SpeechTEK2008*, D102 – Speaking and Listening to Mobile Devices
25. Yuschik, M. (2008c) Multimodal Agent-Mediated Call Center Services, *Voice Search 2008*, San Diego, CA
26. Yuschik, M. (2009) Call Center Multimodal Voice Search, *Voice Search 2009*, San Diego, CA
27. Yuschik, M. (2010) in Meisel, W., Speech In the User Interface: Lessons from Experiences, TMA Publications, Tarzana, CA

Chapter 7

“How am I Doing?”: A New Framework to Effectively Measure the Performance of Automated Customer Care Contact Centers

**David Suendermann, Jackson Liscombe, Roberto Pieraccini,
and Keelan Evanini**

Abstract Satisfying callers’ goals and expectations is the primary objective of every customer care contact center. However, quantifying how successfully interactive voice response (IVR) systems satisfy callers’ goals and expectations has historically proven to be a most difficult task. Such difficulties in assessing automated customer care contact centers can be traced to two assumptions made by most stakeholders in the call center industry:

1. Performance can be effectively measured by deriving statistics from call logs; and
2. The overall performance of an IVR can be expressed by a single numeric value.

This chapter introduces an IVR assessment framework which confronts these misguided assumptions head on and shows how they can be overcome. Our new framework for measuring the performance of IVR-driven call centers incorporates objective and subjective measures. Using the concepts of *hidden* and *observable* measures, we demonstrate in this chapter how it is possible to produce reliable and meaningful performance metrics which provide insights into multiple aspects of IVR performance.

Keywords Spoken dialog systems • Subjective and objective measures • Hidden and observable measures • Caller Experience • Caller Cooperation • Caller Experience Index

7.1 Introduction

“The customer is king” has been business’s maxim since the launch of capitalism and the free-market economy. Today, most large companies handle a substantial amount of their customer care through telephony-based customer care contact

D. Suendermann (✉)
Principal Speech Scientist, SpeechCycle, Inc.,
26 Broadway, 11th Floor, New York, NY 10004, USA
e-mail: david@speechcycle.com

centers, popularly known as call centers. Traditionally, as companies increase in size, their call volume increases likewise. In fact, quite a few companies have to answer millions of customer service calls per month, or even per week. With such a high call volume, it is crucial to the company's business to be able to assess to what extent the customers' goals and expectations are met, even as these customers are distributed throughout a network of call centers. Further questions also arise, such as how call center performance may vary at different times of the day, during certain days of the week, or at particular times of the year, such as during promotional campaigns. One must also be cognizant of the fact that contact centers serve a diverse demographic mix of callers that contact customer support for a variety of reasons.

In cases where human agents are involved, these questions may potentially be answered, since the agent can monitor how well the customer's goals are being met during the course of each call. Whenever a customer is dissatisfied, the agent can make a record of the nature of the problem and escalate the call to a supervisor. However, the picture changes substantially once the human agent is removed from the equation in automated call centers that support interactive voice response (IVR) platforms. In these implementations, there is often great uncertainty about how the system is performing. However, due to their cost effectiveness, such automated systems (spoken dialog systems) increasingly replace customer service representatives in a multitude of tasks. For example:

1. call routing [6];
2. troubleshooting [1];
3. phone banking [12];
4. stock trading [15]; and
5. travel scheduling [20]

In contrast to human agents, such spoken dialog systems deployed in call centers are fundamentally designed to strictly follow the call center's business logic. Consequently, they are not able to reliably handle unexpected events that may occur during the call.¹

7.1.1 Spoken Dialog Systems

Spoken dialog systems are among the most widely adopted applications of speech and spoken language processing. A spoken dialog system is defined as an application in which a machine communicates with humans using speech [14]. In its simplest form, a spoken dialog system can be described by the functional diagram of Fig. 7.1.

¹During a side conversation at the AVIxD workshop held August 2009 in New York, Mark Stallings reported about a GI stationed in Iraq that called an American IVR while rounds of explosions tore through the air.

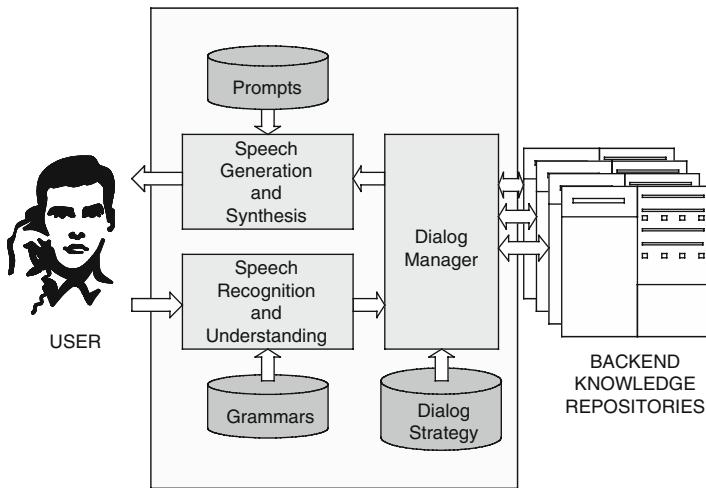


Fig. 7.1 A high-level functional diagram of a spoken dialog system

First, the user’s utterance is processed by a speech recognition and understanding module. Next, the recognized utterance is passed to the dialog manager which uses a set of rules to direct a speech generation module about what information or request will be spoken in response. Finally, this information is sent to a speech synthesizer or prompt player to produce the acoustic output. During this process, the dialog manager may interact with external backend knowledge repositories in order to extract additional information necessary to complete the interaction.

7.1.2 Measures

Two types of measures become relevant for determining the quality of customer service care delivered through an IVR: objective and subjective. In accordance with the definitions of these words [13], objective measures deal “with facts ... as perceived without distortion by personal feelings, prejudices, or interpretations.”

That is, such a measure will always produce the same result, regardless of *who* obtains it or *where* it is obtained. Subjective measures, on the other hand, are based on the measurer’s *own* judgment, and can thus change depending on the measurement conditions.

Objective measures can be further subdivided into *observable* and *hidden* measures. These two terms have a long-standing history in speech and spoken language processing as used, for instance, by hidden Markov models [19] or partially observable Markov decision processes [27]. Here, “observable” means that the “facts” are directly available to the system. Typical observable facts for customer support calls include call duration, whether the user hung up, or how often the system rejected a speech recognition hypothesis. “Hidden,” on the other hand, refers to facts

the system does not know or cannot be certain about, such as how often a caller's utterance failed to trigger the speech recognizer or how often the dialog system selected an incorrect path in the call flow due to a recognition error. Consequently, what we refer to as *hidden measures* are measures of hidden facts that can only be uncovered with certainty through off-line actions as transcription or annotation.

As simple as this sounds, there are many cases where the distinction between observable and hidden measures is not transparent. Consider, for instance, the number of times a caller may have requested to speak to a live agent during the entire call. While one can easily scan the call logs to count how often the speech understanding module hypothesized that the caller requested a human agent (an observable fact) this does not necessarily mean that the caller actually asked that many (or few) times for an agent. For example, the caller may have requested an agent *five* times in a row before the system finally understood the frustrated caller's request for a live agent. In such case, the call log will show that the caller asked for an agent only once during the call, when in reality the caller made five successive agent requests. On the other hand, a nonspeech sound, such as a cough, may have triggered a recognition event that was falsely interpreted by the IVR system as an agent request, when indeed the caller had not made such a request at all. We refer to this type of over-sensitivity in the recognition of agent requests that had not been made as *operator greediness*.²

In normal business operations, hidden facts are often mistakenly treated as observable ones. That is, analysts simply extract statistics about agent requests, distributions on the kinds of issues and concerns customers call about, what they are saying in certain recognition contexts, or the number of speech recognition errors in a call. They do this by processing the call logs to glean such statistics that are treated de facto as if they were "observable" facts. But, in reality, these are all hidden facts and cannot be determined with certainty through automatic means.

It is possible, however, to retrieve observable facts for such cases with manual effort. For example, human beings can listen to the utterances that triggered agent requests to determine whether the recognized requests were real or due to operator greediness. Additionally, utterances collected in recognition contexts can be transcribed to demonstrate what people really said as opposed to what the speech recognizer hypothesized, as well as how often the speech understanding module erred.³ Likewise, cases where the system ignored speech from the caller can also be detected by humans listening to calls. While these types of human annotations

²To make things even more complicated, there are cases where the distinction between observable and hidden becomes fuzzy: Dialog systems may acknowledge that they do not know the facts with certainty and, therefore, work with beliefs, i.e., with probability distributions over the observable facts. For instance, a system may ask a caller for his first name, but instead of accepting the first best hypothesis of the speech recognizer (e.g., "Bob"), it keeps the entire n -best list and the associated confidence scores (e.g., "Bob": 50%; "Rob": 20%; "Snob": 10%, etc.). Such spoken dialog systems are referred to as belief systems [2, 28, 29].

³This can be crucial considering that speech recognition applied to real-world spoken dialog systems can produce word error rates of 30% or higher even after careful tuning [5].

are time consuming and expensive, they are necessary to accurately assess the true performance of a dialog system.

The remainder of this chapter will give an overview of a typical industrial framework designed to perform large-scale objective and subjective analysis (Sect. 7.2) before discussing the most commonly used objective measures (both observable and hidden) applied to assessing dialog system performance (Sect. 7.3). Next, we will introduce two subjective measures, Caller Cooperation and Caller Experience, which we have found to be useful diagnostics to complement the objective measures (Sect. 7.4). Subsequent to that, we will show how the objective measures are related to the subjective ones (Sect. 7.5), and how the latter can be predicted by the former, thus expanding the number of calls that can be analyzed from thousands to millions (Sect. 7.6). Finally, we will conclude by making the case that using a single metric to describe the performance of an arbitrary IVR system (often referred to as the Caller Experience Index) is an untenable solution for accurately measuring automated call center performance (Sect. 7.7).

7.2 Infrastructure

In order to obtain the multiple measurements mentioned in the previous section, many different types of information must be extracted from IVR production calls. In this section, we describe an example infrastructure with all the components necessary to produce the number of objective and subjective measures discussed in more detail throughout the following sections. A functional diagram of the major components of this example infrastructure is shown in Fig. 7.2:

The dialog manager, which controls the interaction, typically resides on a number of application servers (see Fig. 7.1) including the dialog strategy and the integration software that exchanges data with the external backends. Moreover, the dialog manager communicates with the speech processing components such as automatic speech recognition (ASR) and synthesis. In most current industrial implementations, the speech processing components are managed by a voice browser that is the voice analog of a visual web browser. The voice browser receives a VoiceXML, or VXML, page [11] from the application server defining, for example, a prompt to be played to the caller or a grammar⁴ to be used by the speech recognizer for processing the caller’s speech input.

The voice browser saves log entries for each call in local storage. Periodically, batches of log entries are uploaded into databases hosted in the VXML/ASR data warehouse. These entries include, for instance:

- a unique call identifier
- the name and location of all grammars active in the recognition context
- the n best recognition hypotheses

⁴ See Sect. 7.3 for more details on grammars used in IVRs.

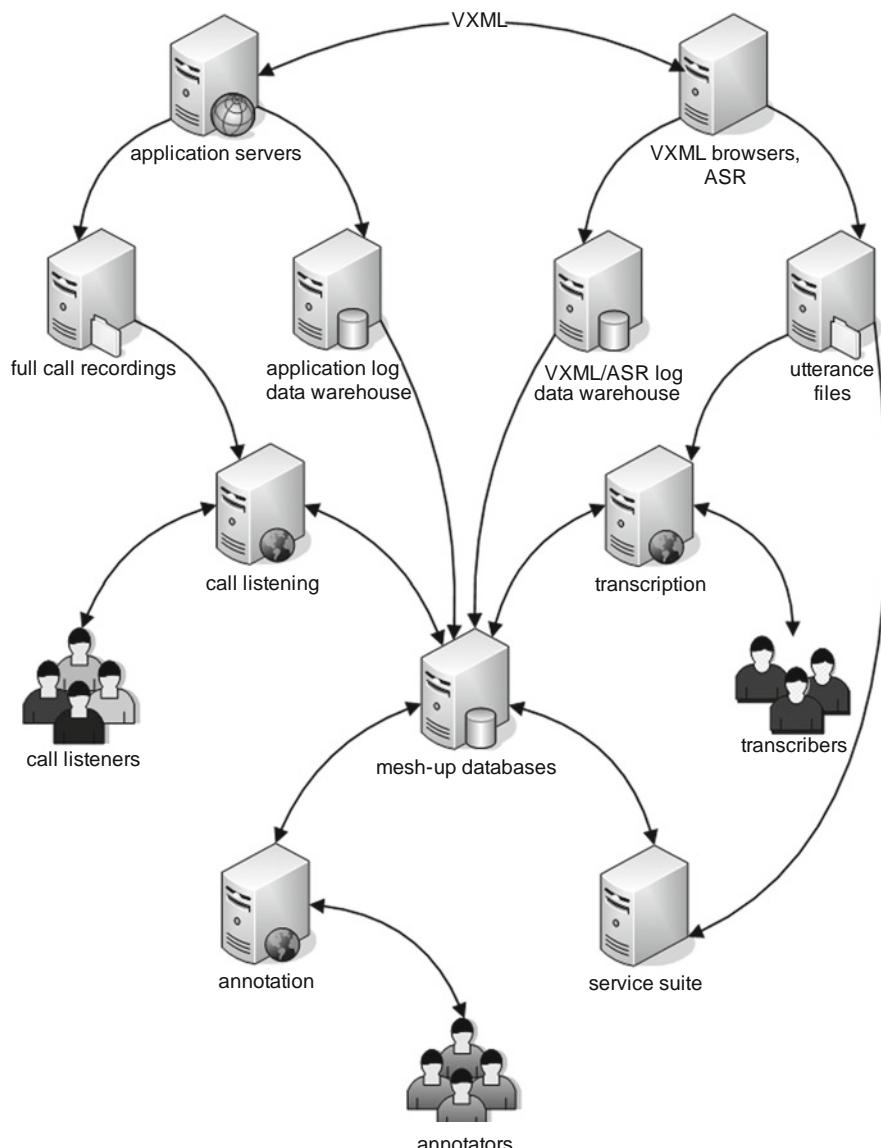


Fig. 7.2 Example of an IVR assessment infrastructure

- the confidence scores for the n best recognition hypotheses
- the m best semantic categories
- the event time
- activity names (i.e., recognition context name)
- name and location of the recorded audio files (explained below)

In addition to the log entries, the speech recognizer can also store a file for each chunk of audio (usually a caller utterance) processed by the ASR. These files are stored on an utterance file server.

Similar to the voice browser, the application servers also store log information in a number of databases hosted in an application log data warehouse. These log entries include, for instance:

- a unique call identifier
- dialog activity names
- activity outbound transitions
- runtime exceptions
- variables used by the application such as data retrieved from backend integration
- reporting variables
- the event time
- the name and location of the recorded full-duplex call audio files (see next item)

The application server is also able to store a full-duplex recording of the entire call including the caller’s input speech and touch tone events, the IVR’s speech, hold music, human agent interventions, etc. These files are stored on a full call recording server.

A subset of the data available in the application log data warehouse and the VXML/ASR data warehouse is copied onto a server which hosts the databases where the multiple sets of log data are synchronized and prepared for several types of human processing. We refer to these databases as *mesh-up databases*.

First, the call listening web server receives an assignment of a number of calls specified in the mesh-up databases and displays the calls in a web interface such as the one shown in Fig. 7.3. The underlying recordings are made available by connecting to the full call recording server. Human call listeners then assess certain aspects of the full call recordings according to principles that will be defined in Sect. 7.4.

The results of this assessment are sent back via the call listening web server and written to the mesh-up databases. A group of human transcribers and annotators are then instructed to use an application such as the one shown in Fig. 7.4 to transcribe and annotate a number of utterances. These utterances are provided by the transcription and annotation servers that expose audio files stored on the utterance file server as well as utterance properties stored in the mesh-up databases. Transcriptions and annotations are written back to the mesh-up databases via the transcription and annotation servers.

Finally, the service suite, hosted on one or several servers, accesses the mesh-up databases as well as the underlying utterance audio files to perform the following automatic services:

- transcription and annotation quality assurance (see [25] for details)
- automatic annotation and transcription (see [24] for details)
- objective performance analysis (see Sect. 7.3)

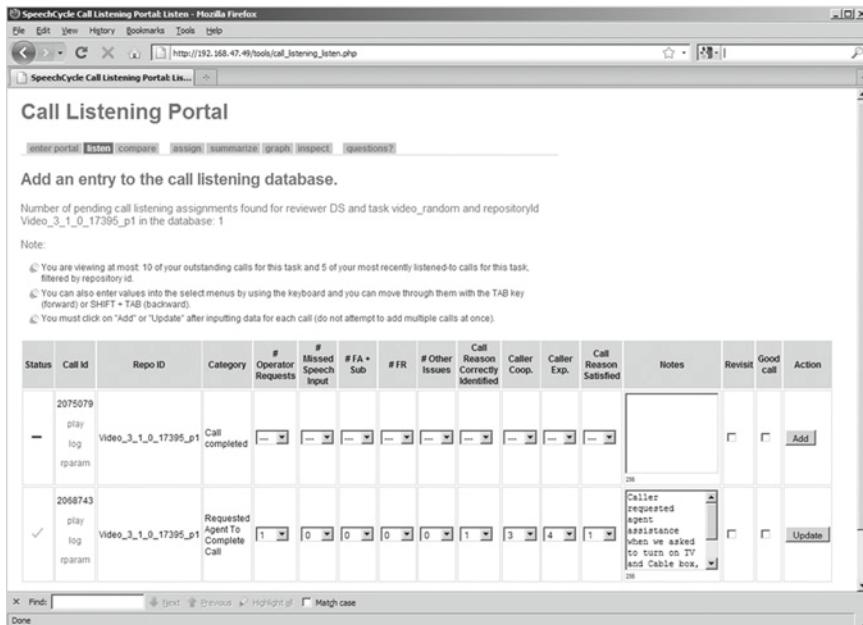


Fig. 7.3 Example of a call listening web interface

| Transcribed Text | isAudioAvailable | Annotated Value | Conf. | Recognized Text | Recog. |
|---|------------------|-------------------------------------|-------|------------------------------------|--------|
| replacement modem | yes | HST_Problem_modem | 600 | replace my modem | Searc |
| internet problems | yes | HST_Problem_Proble | 740 | internet problems | Searc |
| representative | yes | Search_Service_Repres | 500 | operator | Searc |
| representative | yes | operator | 540 | representative | Searc |
| customer service complaint | yes | Search_Complaint | 680 | customer service complaint | oper |
| representative | yes | operator | 490 | representative | oper |
| representative | yes | operator | 570 | representative | oper |
| speak to representative help | yes | operator | 670 | speak to representative help | oper |
| customer service | yes | operator | 500 | customer service | oper |
| customer service | yes | operator | 820 | customer service | oper |
| make a payment | yes | Search_Account_Bill_MakePayment | 440 | make operator | oper |
| make a payment | yes | operator | 490 | make a payment | oper |
| representative | yes | Search_Account_Bill_MakePayment | 490 | representative | oper |
| no my telephone is out its been out for bi... | yes | Phone_Other_Broken | 430 | no my telephone and file has b... | Phoni |
| i want my telephone fixed [2] | yes | Phone_Other_Broken | 650 | i want my telephone fixe | Phoni |
| talk to someone about my phone service | yes | Phone_Other_Vague | 540 | talk someone down my phone s... | Phoni |
| i want order new services | yes | Search_Account_Bill_NewServices | 500 | order new services | Phoni |
| my voice mail is not coming through red ligh... | yes | Phone_VoiceMail | 700 | my voicemail is not coming thro... | Phoni |
| ah balance | yes | Search_Account_Bill_Balance | 620 | uhh balance | Searc |
| i want to know the location | yes | Search_Account_Bill_Center_Locate | 690 | i want the location | Searc |
| payment | yes | Search_Account_Bill_MakePayment | 700 | payment | Searc |
| pay my bill | yes | Search_Account_Bill_MakePayment | 700 | pay my bill | Searc |
| [2] to make a pay | yes | Search_Account_Bill_MakePayment | 360 | make a payment | Searc |
| pay my bill | yes | Search_Account_Bill_MakePayment | 660 | pay my bill | Searc |
| wanna make a payment | yes | Search_Account_Bill_MakePayment | 370 | wanna payment | Searc |
| make a payment | yes | Search_Account_Bill_MakePayment | 420 | make a payment | Searc |
| pay | yes | Search_Account_Bill_MakePayment | 430 | pay | Searc |
| payment | yes | Search_Account_Bill_MakePayment | 450 | payment | Searc |
| pay my bill | yes | Search_Account_Bill_MakePayment | 460 | pay my bill | Searc |
| payment | yes | Search_Account_Bill_MakePayment | 730 | payment | Searc |
| pay my bill | yes | Search_Account_Bill_MakePayment | 730 | pay my bill | Searc |
| make a payment | yes | Search_Account_Bill_MakePayment | 500 | a payment | Searc |
| pb password problem | yes | Search_Account_Bill_MakePayment | 470 | pb pay | Searc |
| parental controls | yes | Search_Account_Bill_MakePayment | 800 | payment | Searc |
| scrolling numbers | yes | Search_Account_Bill_MakePayment | 800 | payment | Searc |
| clock | yes | Search_Account_Bill_PaymentArran... | 450 | arrangement | Searc |

Fig. 7.4 Example of a transcription and annotation software

- subjective performance analysis (see Sect. 7.4)
- prediction of Caller Experience (see Sect. 7.6)

7.3 Objective Measures

7.3.1 Observable Measures

As stated in Sect. 7.1, observable measures are defined as those that the system can be certain about without any additional external information. In the section below, we discuss at length the most common observable measures used to assess the performance of deployed spoken dialog systems.

7.3.1.1 Automation Rate

The automation rate (a.k.a. deflection rate, completion rate) is by far the most important metric used in spoken dialog systems. It measures the percentage of calls in which the caller’s objective was satisfied. Such objectives may include, for example:

- The percentage of calls to a troubleshooting application that ended with the problem being resolved.
- The percentage of calls in a call router that ended up with the right type of agent or IVR.
- The percentage of calls in a bus scheduling application that provided the requested information.

The criteria for labeling a call as successfully automated, however, are diverse and most often depend on business requirements. This is because the cost savings produced by commercial dialog systems are mostly estimated based on automation rate. For example, a call in a high-speed Internet troubleshooting application may be considered automated when the system receives a positive confirmation from the caller that the problem is solved. However, a large number of callers are not patient enough to wait to answer the final confirmation question, but rather hang up on the system as soon as the problem is resolved. Consequently, some calls where the caller hangs up should be considered *successfully* automated. On the other hand, callers may simply hang up on the system out of frustration. So, an analyst may want to consider the specific context in which callers hang up to determine whether calls were successfully automated or not.

To make things yet more complicated, even some calls that include customer confirmations at the end can be problematic. Consider, for instance, the following exchange:

System: Just to confirm, you are able to connect to the Internet now, is that right?
Caller: What?

It is possible for the speech recognizer to falsely interpret the caller's input to this prompt as "yes" instead of "what." In this case, the system would then label the call as successfully automated and may terminate the call. Because of problematic cases like these where it is impossible for the system to establish the truth with certainty, automation rate could possibly be considered a hidden measure as well. However, based on the origin of the term *automation*, which means that a human's task was performed by a machine, we can equally regard a call as automated whenever no human agent was necessary for the completion of the task. Regardless of whether the system hangs up on the caller or the caller hangs up on the system, both scenarios do not involve a human agent, and the call can be considered automated if the same caller does not call back for the same issue within a specified amount of time (typically 24 h). Since this fact is *known* by the system, we categorize automation rate as an observable measure despite any possible ambiguity, as described above.

7.3.1.2 Average Handling Time

Average Handling Time (AHT) is the average duration of calls handled by an IVR. Companies with deployed dialog systems often try to minimize AHT because it is linearly correlated with data hosting costs.

As a simple example, consider a spoken dialog system handling one million calls per month with an AHT of 5 min. Also, let us assume that the volume of calls handled by the system is constant at all times (which is no doubt far from true for real-world systems). Let us further assume that an application server and a VXML browser/ASR server can each handle ten calls at a time. Based on these assumptions, we can calculate that the minimum hardware requirements for this system would consist of 12 application and 12 VXML browser/ASR servers. Now, reducing AHT by 1 min would enable the elimination of two application and two VXML browser/ASR servers. This would undoubtedly lead to a considerable cost reduction in hardware and licensing fees. Another important reason to minimize AHT is its effect on the Caller Experience (see Sect. 7.4). Everything else being equal, longer calls may be annoying to most callers, and thus considerably reduce the Caller Experience.

7.3.1.3 “Speech Errors” and Retry Rate

As we will see in Sect. 7.3.2, speech recognizers can treat speech input in several ways. They may directly accept the input, attempt to confirm it with the caller, or reject it in the case of uncertainty or when the input is clearly out of the scope of the

recognition context. Furthermore, in response to a confirmation prompt, the caller may confirm, dis-confirm, or reply once more with an utterance that the recognizer is not certain about or deems out of scope. Recognizer rejections, caller dis-confirmed, and other types of exceptional conditions such as time-outs or speech overflows are often considered together as “speech errors.” In many cases, they are not actual errors as, for instance, when a caller’s coughing is rejected or when the caller remains silent and that results in a time-out. To identify events as real errors, one needs to assess hidden measures as described below, because one needs to know the actual caller input rather than the system’s hypothesis of the input.

Without the capacity or infrastructure to evaluate hidden measures (i.e., when no transcriptions or annotations are available), the performance of the speech recognition and understanding system is often estimated by means of the retry rate. This metric is the average number of turns it takes to gather a piece of information from the caller minus one. The inclusion of minus one in the metric is due to the fact that an optimal exchange of information takes only one turn, and, thus, has zero retries. Under this definition, it is not quite clear whether confirmations should be considered as turns. On the one hand, it could be argued that confirmations reflect poor recognizer performance, and can also be annoying if applied aggressively. Following this logic, confirmations should be reflected in the retry rate. Consider the following unsuccessful dialog that takes this notion to an absurd extreme:

System: How many lights are blinking?
Caller: Three.
System: Did you say *three*?
Caller: Yes.
System: You just said *yes*, right?
Caller: Right.
System: You just said *right*, right?
...

On the other hand, if a system designer desires to reduce the retry rate as much as possible, a trivial solution would be to accept everything: no rejections and no confirmations. The resulting retry rate would always be zero. However, the performance of this system would be far from ideal. In this case, subjective measures, such as those described in Sect. 7.4, would be necessary to provide a complete assessment of the system’s performance.

7.3.1.4 Hang-Ups and Opt-Outs

Considering that we have just determined that retry rate and the number of speech errors are unreliable observable measures, how can one possibly assess an IVR’s quality by means of observable measures other than automation rate? How can we obtain an idea of how the callers feel about the system and the quality of the interaction? Two additional indicators of potential problems are when callers request

human-agent assistance (opt-out) or hang up before the call has been completed. However, these measures too may actually not be observable in that a system may think callers requested an agent because of an ASR error even though they actually did not, an effect we earlier referred to as operator greediness.

7.3.2 *Hidden Measures*

Most of the speech recognition contexts in commercial spoken dialog systems are designed to map the caller's input to one of a finite set of context-specific semantic classes [9]. This is done by providing a grammar for the speech recognizer at every given recognition context. A grammar serves two purposes:

1. It constrains the lexical content the recognizer is able to recognize in this context (the language model).
2. It assigns one out of a set of possible semantic classes to the recognition hypothesis (the classifier).

Acoustic events processed by spoken dialog systems are usually split into two main categories: In-Grammar and Out-of-Grammar. In-Grammar utterances are those that belong to one of the semantic classes that can be processed by the system logic in the given context. Out-of-Grammar utterances comprise all remaining events, such as utterances whose meanings are not handled by the grammar or when the input is nonspeech noise.

Spoken dialog systems usually respond to acoustic events that were processed by a grammar in one of three ways:

1. The event is rejected. This is when the system either assumes that the event was Out-of-Grammar or has such a low confidence value for its In-Grammar semantic class that it rejects the utterance. In such cases, callers are usually re-prompted for their input.
2. The event is accepted. This is when the system detected an In-Grammar semantic class with high confidence.
3. The event is confirmed. This is when the ASR assumes that it correctly detected an In-Grammar semantic class with a low confidence. Consequently, the caller is asked to verify the predicted class. Historically, confirmations have not been used in those contexts where they would potentially confuse the caller, for instance in yes/no contexts (see the above example dialog on retry rate).

Based on these categories, an acoustic event and the system's corresponding response can be described by four binary questions:

1. Is the event In-Grammar?
2. Is the event accepted?
3. Is the event correctly classified?
4. Is the event confirmed?

Table 7.1 In-Grammar? Accepted?

| | A | R |
|---|----|----|
| I | TA | FR |
| O | FA | TR |

Table 7.2 Event acronyms

| | |
|------|-----------------------------|
| I | In-Grammar |
| O | Out-of-Grammar |
| A | Accept |
| R | Reject |
| C | Correct |
| W | Wrong |
| Y | Confirm |
| N | Not-Confirm |
| TA | True Accept |
| FA | False Accept |
| TR | True Reject |
| FR | False Reject |
| TAC | True Accept Correct |
| TAW | True Accept Wrong |
| FRC | False Reject Correct |
| FRW | False Reject Wrong |
| FAC | False Accept Confirm |
| FAA | False Accept Accept |
| TACC | True Accept Correct Confirm |
| TACA | True Accept Correct Accept |
| TAWC | True Accept Wrong Confirm |
| TAWA | True Accept Wrong Accept |
| TT | True Total |
| TCT | True Confirm Total |

Now, we can draw a diagram containing all possible combinations of outcomes to the first two questions as shown in Table 7.1. (Abbreviations for all acoustic event classification types used in this chapter are presented in Table 7.2.)

The third question is only relevant for In-Grammar events, since Out-of-Grammar utterances comprise a single class, and can therefore only be either falsely accepted or correctly rejected. The corresponding diagram for all possible outcomes to the first three questions is thus shown in Table 7.3.

Finally, extending the diagram to accommodate the fourth question about whether a recognized class was confirmed is similarly only relevant for accepted utterances, as rejections are never confirmed; see Table 7.4.

When the performance of a given recognition context is to be measured, the analyst can collect a certain number of utterances recorded in this context, look at the recognition and application logs to see whether these utterances were accepted or confirmed and which class they were assigned to, transcribe and annotate the utterances according to their true semantic class and, finally, count the events and divide them by the total number of utterances. If X is an event from the list in Table 7.2, we

Table 7.3 In-Grammar? Accepted? Correct?

| | A | | R | |
|---|-----|-----|-----|-----|
| | C | W | C | W |
| I | TAC | TAW | FRC | FRW |
| O | FA | | TR | |

Table 7.4 In-Grammar? Accepted? Correct? Confirmed?

| | A | | R | |
|---|---|------|------|-----|
| | C | W | C | W |
| I | Y | TACC | TAWC | FRC |
| | N | TACA | TAWA | |
| O | Y | FAC | | TR |
| | N | FAA | | |

will refer to x as this average score, e.g., tac is the fraction of total events correctly accepted.

In order to report system recognition and understanding performance concisely, the multitude of measurements described above can be consolidated into a single metric by splitting the events into two groups: good and bad. The resulting consolidated metric is then the sum of all good (hence, an overall accuracy) or the sum of all bad events (overall error rate). In Tables 7.3 and 7.4, the good events are highlighted. Accordingly, two consolidated summary metrics True Total (tt) and True Confirm Total (tct) are defined as follows [23]:

$$tt = tac + tr \quad (7.1)$$

$$tct = taca + tawc + fac + tr \quad (7.2)$$

In the special case that a recognition context never confirms, (7.2) equals (7.1) since the confirmation terms $tawc$ and fac disappear and $taca$ becomes tac (due to the fact that $tacc$ is zero).

7.4 Subjective Measures

The objective measures discussed in the previous section are able to shed light on most of the aspects normally considered in assessing the performance of spoken dialog systems. We know whether tasks are completed (automation rate), whether we do an efficient job (AHT), whether the callers cooperate with the system (hang-ups, opt-outs), and whether the speech recognition and understanding performance are state-of-the-art (True Total). What more could we possibly need to accurately evaluate an IVR system? As an example of the deficiency of using these measures alone, consider the following dialog:

system: What are you calling about today?

caller: I lost my password.

system: Sorry, I cannot help with your password.
caller: Agent.
system: <hold music>
caller: Agent!!!
system: Goodbye. <system hangs up>

This call may be considered automated as the system delivered a message and hung up. The call is short and features very few dialog turns. In fact, it only registered one user input. The agent requests were completely ignored and thus were likely not even reported in the application or VXML browser/ASR logs. As the system neither heard nor reported on the agent requests and the system itself hung up, both opt-out and hang-up counts of this call were zero. Finally, the single user input was correctly accepted as a password-related utterance resulting in a True Total of 100%. Thus, an assessment using only these objective measures would describe the system’s performance as perfect.

This, however, would be a gross error. What happened to the “primary goal of every customer care contact center” as stated in the abstract of this chapter? Did the system actually “satisfy the caller’s goals and expectations”? By no means! The caller wanted help with his password, but did not receive it. The caller requested a human agent, but there was no response from the system. The caller *demanded* a human agent, and again received no response.

Thus, to quantify the fulfillment of the system’s primary goal when objective measures do not suffice, we need to make use of subjective measures. This enables the analyst to detect cases where the system contains logical flaws, gathers redundant or irrelevant information, ignores the caller’s speech, goes down the wrong dialog path, or simply sounds terrible.

The subjective measures that have commonly been used in spoken dialog systems require callers to respond to a survey about their experience with the dialog system and contain questions such as the following [8, 3, 17, 21]:

- Did you complete the task?
- Was the system easy to understand?
- Did the system understand what you said?
- Did the system work the way you expected it to?

However, this type of subjective data is not necessarily reliable, due to the fact that different users may interpret the questions differently. Furthermore, little empirical research has been done into the selection of the specific questions contained in the survey [7]. Finally, such surveys are not practical in a real-time system deployed in a commercial setting because participation in the survey must be optional; consequently, any data collected from the survey would represent a skewed sample of callers.

Suppose, for instance, that we want to measure to what degree the system’s “primary goal” was fulfilled on a five-point scale, and that we do this by asking a number of callers after the call for a rating and averaging over these ratings. Now, in doing so, we can obtain an average performance score as was the case for the objective measures. However, this score’s reliability is questionable, and the degree of questionability is best manifested by the high variance of the subjective ratings.

Now, one may argue that this high variance may be due to the diversity of the calls themselves: there may be calls that went perfectly and others that ended up in a disaster. So, to separate the variance inherent to the task from that due to the rater's subjectivity, the optimal approach would be to have several subjects rate the same calls.

This idea obviously requires subjects other than the callers themselves, and thus requires a full-duplex recording of the entire caller-IVR conversation to be retained. Then, a team of expert evaluators listens to the calls and generates multiple ratings for each call. Some advantages of this method in comparison to the conventional survey-based approach include:

- Multiple ratings of the same call are possible.
- Ratings are independent of the caller's emotional state and, hence, more reliable.
- Ratings are available for a call even though the caller may not have been willing to participate in a survey.
- Evaluators can be trained. Thus, they can have a deep understanding of the system's functionality, purpose, components, features, and limitations. In fact, in the optimal case, the evaluator team includes the people who built the original IVR such as voice user interface designers, speech scientists, and application engineers.
- The rating is independent of the time the call was made. While a caller survey can only be completed within a short window of time after the termination of the call (in order to ensure that the details of the call are fresh in the caller's memory), evaluators can rate any given call months after it was recorded. This is a crucial point, since it is often useful to produce ratings of a certain call-type population after the fact. For example, it may turn out that the hang-up rate doubled after a new release of an application. In this case, a call listening project would focus on calls where callers hung up in certain situations. Furthermore, results of a new release may be compared to an older version of the same system where the hang-up rate was lower, including calls that were recorded months earlier.

Depending on the objective of a call listening project, a variety of rather specific call properties can be explored, such as

- How often input from the caller was ignored.
- Whether logical flaws occurred in the call.
- Whether the call reason was correctly identified.
- Whether the call was routed to the correct place.
- Any additional issues that the evaluator noticed.
- Caller Cooperation and Caller Experience (more details on these metrics will be provided in the following section).

Returning to the system's “primary goal,” one can ask why we do not simply include a single rating for the “satisf[action of the] caller[’s] goals and expectations.” The reason is that we found that the caller's goals and expectations can only be satisfied when the following two conditions are met:

1. The spoken dialog system does a good job.
2. The caller does a good job.

If the caller does not want to respond, does not listen, hangs up, opts out, calls with goals entirely out of the system’s focus, etc., then the system will likely fail regardless of how well it was designed. This type of failure, however, is not indicative of the IVR’s actual capability. To account for this observation and to separate the caller’s contribution to the outcome of a call from the system’s contribution, we introduced the concepts of Caller Cooperation and Caller Experience.

7.4.1 Caller Cooperation

Caller Cooperation is a qualitative measure of the caller’s willingness to participate in a conversation with a spoken dialog system. This measure is a rating on a discrete five-point scale, where 1 indicates no cooperation and 5 indicates full cooperation. For example, “belligerent” operator requests and the caller ignoring all prompts warrants a rating of 1. On the other hand, if callers always respond cooperatively to the actual conversational context, the call is given a caller cooperation score of 5. Take, for example, a situation in which callers are asked whether they want to interact with the system or whether they want to speak with an agent. Even if they answer “no” to the former and “yes” to the latter they are still considered fully cooperative because both of those answers respond to the question.

7.4.2 Caller Experience

Caller Experience measures how well the system treats the user. It is also measured on a discrete five-point scale, with 1 for the worst experience and 5 for an optimal experience. Caller Experience is a rating of the entire system design, not just speech recognition and understanding performance. The following are some relevant questions captured by this measure: Were the prompts clear? Were the transitions appropriate? Was the caller helped to achieve intended tasks efficiently? Did the system appear intelligent by using back-end integration whenever possible?

7.5 On the Relationship Between Subjective and Objective Measures

This section reports on a number of experiments concerning the correlation between subjective and objective measures. Specifically, we experimentally investigated the relationship between Caller Experience and a selection of the hidden measures introduced in Sect. 7.3.

7.5.1 Study 1. On the Correlation Between True Total and Caller Experience

In the first of our case studies, the correlation between Caller Experience and hidden measures was quantified. For this purpose, we selected 446 calls from four different spoken dialog systems deployed on the customer service hotlines of three major cable service providers. The spoken dialog systems consisted of

- a call routing application – cf. [22],
- a cable TV troubleshooting application,
- a broadband Internet troubleshooting application, and
- a Voice-over-IP troubleshooting application – see for instance [1].

The calls were evaluated by voice user interface experts, and Caller Experience was rated according to the definition provided in Sect. 7.4. Furthermore, all speech recognition utterances (4,480) were transcribed and annotated with their semantic classes.

Thereafter, we computed all of the hidden measures introduced in Sect. 7.3, and averaged them for the five distinct values of Caller Experience. As an example, Fig. 7.5 shows how the relationship between the mean True Total value and Caller Experience is nearly linear. Applying the Pearson correlation coefficient to this five-point curve yields $r = 0.972$ and confirms that what we see is pretty much a straight line. Comparing this value to the coefficients produced by the individual metrics TAC, TAW, FR, FA, and TR as done in Table 7.5, shows that no other line is as straight as the one produced by True Total. This thus indicates that maximizing this value will produce spoken dialog systems with the highest level of user experience.

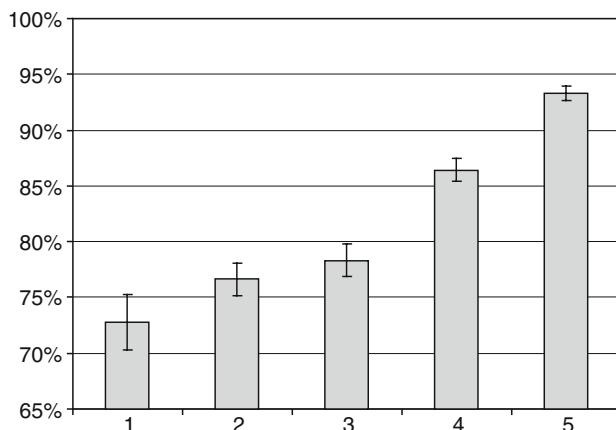


Fig. 7.5 Dependency between Caller Experience and True Total

Table 7.5 Pearson correlation coefficient for several utterance classification metrics after grouping and averaging

| | A | | R |
|---|-------|--------|--------|
| | C | W | |
| I | 0.969 | -0.917 | -0.539 |
| O | | -0.953 | -0.939 |

7.5.2 Study 2. Continuous Tuning of a Spoken Dialog System to Maximize True Total and Its Effect on Caller Experience

The second study presents a practical example of how rigorous improvement of speech recognition and understanding leads to real improvement in the Caller Experience metric.

The dialog system we examined was actually an integration of the four systems listed in the previous section. When callers access the service hotline, they are first asked to briefly describe the reason for their call. After a maximum of two follow-up questions to further disambiguate the reason for their call, they are either connected to a human operator or one of the three automated troubleshooting systems. Escalation from one of these systems can connect the caller to an agent, transfer the caller back to the call router or to one of the other troubleshooting systems.

When the application was launched in June 2008, its True Total averaged 78%. During the following three months, almost 2.2 million utterances were collected, transcribed, and annotated for their semantic classes to train statistical grammars in a continuously running update process [26]. Whenever a grammar significantly outperformed the most recent baseline, it was released and put into production leading to an

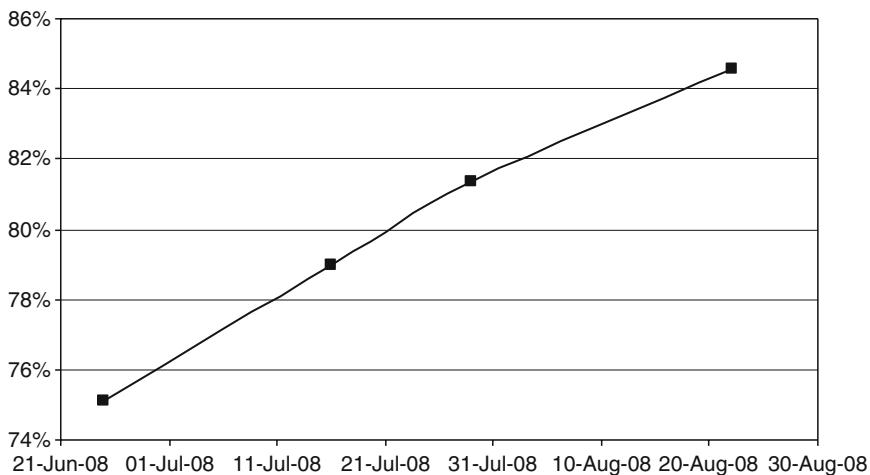


Fig. 7.6 Increase of the True Total of a large-vocabulary grammar with more than 250 classes over release time

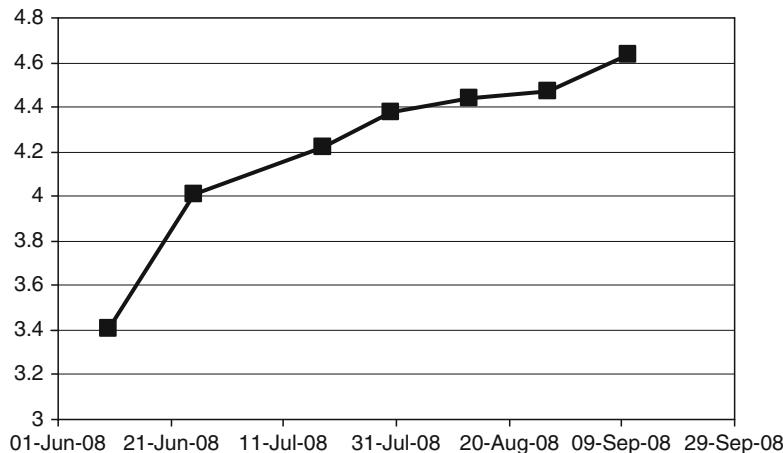


Fig. 7.7 Increase of Caller Experience over release time

incremental improvement of performance throughout the application. As an example, Fig. 7.6 shows the True Total increase of the top-level large-vocabulary grammar that distinguishes more than 250 classes. The overall performance of the application increased to more than 90% True Total within three months of its launch.

Having witnessed a significant gain of a spoken dialog system's True Total value, we would now like to know to what extent this improvement resulted in an increase of Caller Experience. Figure 7.7 shows that Caller Experience did indeed improve substantially. Over the same three-month period, we achieved a monotonic increase from an initial Caller Experience of 3.4 to a final value of 4.6.

7.6 Predicting Subjective Measures Based on Objective Measures

Expert listening is a reliable way to ascertain subjective ratings of Caller Cooperation and Caller Experience. However, there are a number of obvious limitations to this approach:

- Human listening is expensive and does not easily scale. This means that call listening projects are very often limited to some hundreds of calls as compared to millions of calls or utterances that can be processed by an objective analysis as discussed in Sect. 7.3.
- Human listening is time consuming⁵.
- Human listening cannot be done in real time and is therefore not applicable to any live analysis or reporting infrastructure.

⁵"A 19 minute call takes 19 minutes to listen to" is one of ISCA and IEEE fellow Roberto Pieraccini's famous aphorisms.

This raises the question: Can the generation of subjective ratings be automated? The results of our research into the correlation between objective and subjective measures reported in the previous section show that it should be possible, at least to a certain extent.

In this section, we discuss a method of predicting the subjective Caller Experience rating based on objective scores trained using data from 1,500 calls annotated by 15 expert listeners. These calls came from the same call routing application which distinguishes over 250 call categories [22]. Eighty-five percent of these calls served as training data for a classification algorithm, and the remaining 15% were set aside for testing. Each call of the test set was rated by three listeners to see how well the human listeners perform when compared with each other, and how well the classifier performs when compared to human listeners. Below, these three sets of human annotations will be referred to as *human1*, *human2*, and *human3*.

For each call in the training set, a feature vector consisting of multiple observable and hidden objective measures was established. A selection of these features includes:

- True Total
- number of opt-outs
- the classification status of the call (how well the system determined the reason for the call)
- the exit status of the call (whether the caller’s task was completed or where the caller was subsequently transferred)

A decision tree [18] was chosen for the statistical classifier since its model is easy to interpret and can provide useful information about the relative importance of the features in the feature set. For each call in the test set, the classifier chose the most likely class (Caller Experience rating) by following the nodes of the decision tree model corresponding to the feature values for that call. The set of Caller Experience ratings predicted by this classifier are referred to as *auto* below. Further details on the implementation of this experiment can be found in [4].

Finally, the test set ratings from the three sets of human listeners were compared with each other as well as with the predictions made by the classifier. In order to determine how well the different sets of listeners agreed in their subjective evaluation of Caller Experience for each call, Fig. 7.8 shows the frequencies of different levels of rating differences for each human-to-human comparison. The percentages of calls in which the two human listeners agreed completely (i.e., when they provided the exact same Caller Experience rating) are 54.0, 56.9, and 59.4% for the three human-to-human comparisons. Similarly, the combined percentages of calls in which the two human listeners differed by at most one point were 88.7, 87.6, and 91.6%, respectively.

The predictions from the classifier for each call were also compared to the ratings provided by the three sets of human listeners as shown in Fig. 7.9. Surprisingly, the percentage in each set achieving a rating either identical or within one point was on par with that of the human-to-human comparison: 88.1, 95.5, and 92.1%, respectively.

These results indicate that automatic classification of Caller Experience can produce results that are as consistent as human ratings (refer to [4] for more discussion of this matter). This finding means that the main contributions to human subjective

Fig. 7.8 Comparison of agreement among human listeners

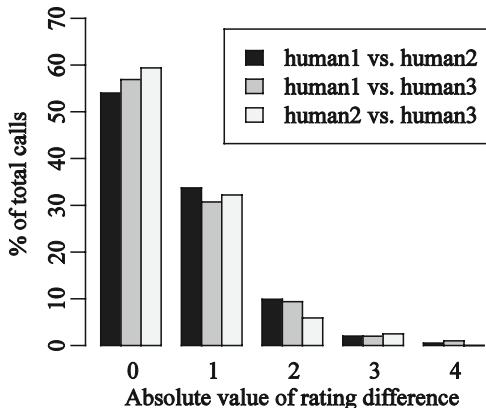
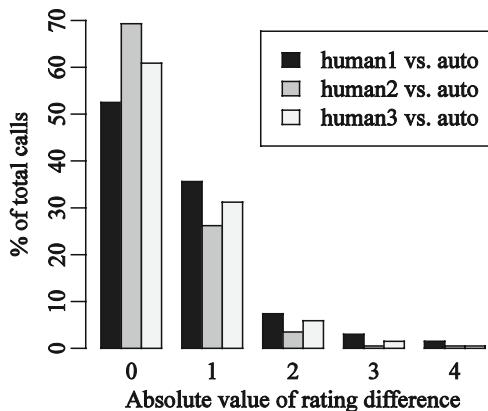


Fig. 7.9 Comparison of agreement between human listeners and classifier



ratings come from call characteristics that are covered by objective measures (the classifier only used feature vectors composed of objective measures as input). Indeed, there are system defects that will not be explicitly covered by the objective measures such as the collection of irrelevant or redundant information from the caller or logical flaws in the interaction (as discussed in the introduction to this section). However, we have observed a considerable correlation between these types of problems and objective measures such as opt-outs or hang-ups thus supporting the robustness of the proposed prediction approach.⁶

⁶To give an example: We recently heard a call where the caller said “Cannot send e-mail” in a call-routing application and was forwarded to an automatic Internet troubleshooting application. This app took care of the problem and supposedly fixed it by successfully walking the caller through the steps of sending an e-mail to himself. Thereafter, the caller was asked whether there was anything else he needed help with, and he said “yes.” He was then connected back to the call router where he was asked to describe the reason for his call, and he said “Cannot send e-mail.” Instead of understanding that the caller’s problem was obviously not fixed by the Internet troubleshooting application during the first turn, he was routed there again and went through the same steps as he did the first time. Eventually, the caller requested human-agent assistance, understanding that he was caught in an infinite loop. Here, the caller’s opt-out was directly related to the app’s logical flaw.

7.7 Searching for the Caller Experience Index

In the previous sections of this chapter, we discussed dozens of measures that can be used to evaluate a spoken dialog system: objective ones, subjective ones, hidden ones, observable ones, ones for confirmation and rejection, speech recognition and understanding, ones for call success, duration, and routing precision, ones to describe how callers are treated by IVRs and how IVRs are treated by callers, and so forth. How can customer care managers, technology vendors, and marketing and sales representatives understand how a system is doing overall? Naturally, there is a high demand in the industry for a single standardized metric to concisely describe system performance, similar to word error rate for an ASR system or the Bilingual Evaluation Understudy (BLEU) score for machine translation technology [16]. Let us call this metric the *Caller Experience Index*.

How do we combine all of the assessment machinery discussed in this chapter into a single number between, say, 1 and 5? Does a score of 4 mean that the system sounds pleasant? That it automates successfully? That it produces optimally short calls and thus saves hosting fees? On the other hand, does a score of 2 mean that the system fails to recognize caller inputs? That callers hang up out of frustration? That something was wrong with the backend integration? It is unclear. Furthermore, are two different systems rated with the same score interchangeable? It is possible that the one does an outstanding job of escalating callers to human agents, thus increasing Caller Experience but failing to automate calls, whereas the other tries hard to automate calls but annoys callers by routinely ignoring their requests for human assistance?

After assessing millions of calls in our search for the elusive Caller Experience Index, we found that, simply said, there cannot be a single number that tells the truth about any given system. Rather, the optimal score depends on the business goals for each specific system. If the only objective is financial, the best score is some combination of automation rate and AHT, thus completely ignoring hidden and subjective measures, i.e., the opinion of the caller (see, e.g., [10]). On the other hand, if the system designer aims for the most pleasant treatment of a preferred customer group, the best implementation would optimize for Caller Experience and minimize speech recognition and understanding problems. Additionally, the application could include a so-called opt-in (i.e., an explicit offer to speak to a live agent at any time) which would artificially boost the opt-out rate. In a heavily trafficked call routing application, the primary goal would be to keep the callers connected to the system until they were successfully routed. This would be achieved by optimizing some combination of opt-out and automation rates. And so on, and so forth.

Useful assessment of spoken dialog systems will therefore remain constrained by the customer’s preferred optimization. For every new application and every new business scenario, the analytic team must agree on the application’s primary objective and accordingly weight and combine some or all of the measures discussed herein (and possibly others not covered in this chapter) to create its own specific and proprietary version of the Caller Experience Index. Hence, to consult

Merriam-Webster for the last time, the idea of a single, widely applicable Caller Experience Index will remain “an unverified story handed down from earlier times” or, in short, a legend.

References

1. Acomb, K., Bloom, J., Dayanidhi, K., Hunter, P., Krogh, P., Levin, E., and Pieraccini, R. (2007). Technical Support Dialog Systems: Issues, Problems, and Solutions. In Proc. of the HLT-NAACL, Rochester, USA.
2. Bohus, D. and Rudnicky, A. (2005). Constructing Accurate Beliefs in Spoken Dialog Systems. In Proc. of the ASRU, San Juan, Puerto Rico.
3. Danieli, M. and Gerbino, E. (1995). Metrics for Evaluating Dialogue Strategies in a Spoken Language System. In Proc. of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, Torino, Italy.
4. Evanini, K., Hunter, P., Liscombe, J., Suendermann, D., Dayanidhi, K., and Pieraccini, R. (2008). Caller Experience: A Method for Evaluating Dialog Systems and Its Automatic Prediction. In Proc. of the SLT, Goa, India.
5. Evanini, K., Suendermann, D., and Pieraccini, R. (2007). Call Classification for Automated Troubleshooting on Large Corpora. In Proc. of the ASRU, Kyoto, Japan.
6. Gorin, A., Riccardi, G., and Wright, J. (1997). How May I Help You? Speech Communication, 23(1/2).
7. Hone, K. and Graham, R. (2000). Towards a Tool for the Subjective Assessment of Speech System Interfaces (SASSI). Natural Language Engineering, 6(34).
8. Kamm, C., Litman, D., and Walker, M. (1998). From Novice to Expert: The Effect of Tutorials on User Expertise with Spoken Dialogue Systems. In Proc. of the ICSLP, Sydney, Australia.
9. Knight, S., Gorrell, G., Rayner, M., Milward, D., Koeling, R., and Lewin, I. (2001). Comparing Grammar-Based and Robust Approaches to Speech Understanding: A Case Study. In Proc. of the Eurospeech, Aalborg, Denmark.
10. Levin, E. and Pieraccini, R. (2006). Value-Based Optimal Decision for Dialog Systems. In Proc. of the SLT, Palm Beach, Aruba.
11. McGlashan, S., Burnett, D., Carter, J., Danielsen, P., Ferrans, J., Hunt, A., Lucas, B., Porter, B., Rehor, K., and Tryphonas, S. (2004). VoiceXML 2.0. W3C Recommendation. <http://www.w3.org/TR/2004/REC-voicexml20-20040316>.
12. Melin, H., Sandell, A., and Ihse, M. (2001). CTT-Bank: A Speech Controlled Telephone Banking System – An Initial Evaluation. Technical report, KTH, Stockholm, Sweden.
13. Merriam-Webster (1998). Merriam-Webster’s Collegiate Dictionary. Merriam-Webster, Springfield, USA.
14. Minker, W. and Bennacef, S. (2004). Speech and Human-Machine Dialog. Springer, New York, USA.
15. Noeth, E., Boros, M., Fischer, J., Gallwitz, F., Haas, J., Huber, R., Niemann, H., Stemmer, G., and Warnke, V. (2001). Research Issues for the Next Generation Spoken Dialogue Systems Revisited. In Proc. of the TSD, Zelezna Ruda, Czech Republic.
16. Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In Proc. of the ACL, Philadelphia, USA.
17. Polifroni, J., Hirschman, L., Seneff, S., and Zue, V. (1992). Experiments in Evaluating Interactive Spoken Language Systems. In Proc. of the DARPA Workshop on Speech and Natural Language, Harriman, USA.
18. Quinlan, J. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco, USA.
19. Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of the IEEE, 77(2).

20. Raux, A., Langner, B., Black, A., and Eskenazi, M. (2005). Let's Go Public! Taking a Spoken Dialog System to the Real World. In Proc. of the Interspeech, Lisbon, Portugal.
21. Shriberg, E., Wade, E., and Prince, P. (1992). Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction. In Proc. of the DARPA Workshop on Speech and Natural Language, Harriman, USA.
22. Suendermann, D., Hunter, P., and Pieraccini, R. (2008a). Call Classification with Hundreds of Classes and Hundred Thousands of Training Utterances and No Target Domain Data. In Proc. of the PIT, Kloster Irsee, Germany.
23. Suendermann, D., Liscombe, J., Dayanidhi, K., and Pieraccini, R. (2009a). A Handsome Set of Metrics to Measure Utterance Classification Performance in Spoken Dialog Systems. In Proc. of the SIGdial Workshop on Discourse and Dialogue, London, UK.
24. Suendermann, D., Liscombe, J., and Pieraccini, R. (2010). How to Drink from a Fire Hose. One Person can Annotate 693 Thousand Utterances in One Month. In Proc. of the SIGDIL, 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Tokyo, Japan.
25. Suendermann, D., Liscombe, J., Evanini, K., Dayanidhi, K., and Pieraccini, R. (2008b). C5. In Proc. of the SLT, Goa, India.
26. Suendermann, D., Liscombe, J., Evanini, K., Dayanidhi, K., and Pieraccini, R. (2009c). From Rule-Based to Statistical Grammars: Continuous Improvement of Large-Scale Spoken Dialog Systems. In Proc. of the ICASSP, Taipei, Taiwan.
27. Williams, J. (2006). Partially Observable Markov Decision Processes for Spoken Dialogue Management. PhD thesis, Cambridge University, Cambridge, UK.
28. Williams, J. (2008). Exploiting the ASR N-Best by Tracking Multiple Dialog State Hypotheses. In Proc. of the Interspeech, Brisbane, Australia.
29. Young, S., Schatzmann, J., Weilhammer, K., and Ye, H. (2007). The Hidden Information State Approach to Dialog Management. In Proc. of the ICASSP, Hawaii, USA.

Chapter 8

“Great Expectations”: Making use of Callers’ Experiences from Everyday Life to Design a Satisfying Speech-only Interface for the Call Center

Stephen Springer

Abstract Speech-activated self-service systems in corporate call centers can provide callers with an experience that is much closer to the ideal of talking with an experienced agent than is possible with TouchTone™ systems. The closer analogy to live conversation has its benefits, but can also introduce pitfalls for the novice designer. In this chapter, we look at the expectations that callers bring to these phone calls, ranging from broad expectations with regard to self-service in general, to the more specific expectations of human-to-human conversation about consumer issues. We recommend several steps to the system designer to produce more successful interaction between callers and speech interfaces. They focus on the thoughtful use of user modeling achieved by employing ideas and concepts related to transparency, choice, and expert advice, all of which most, if not all, callers are already familiar with from their own everyday experiences.

Keywords Speech-only interface • Transparency of real-world self-service systems • Caller expectations • Self-service applications for call centers • Semantic Language Models • Customer satisfaction • Interactive Voice Response (IVR) systems • Call center agent • Positive user experience

8.1 Introduction

For the past decade, one of the most productive areas of speech-activated interface design has been in self-service applications for Call Centers. While earlier self-service applications used TouchTone™ keys, the introduction of speech interfaces, which ask the caller to *speak* instead of *press*, presents far more options to the user. Much of the literature and reported progress in this arena have focused on the science

S. Springer (✉)

Senior Director of User Interface Design, Nuance Communications, Inc.,
1 Wayside Road, Burlington, MA 01803, USA
e-mail: Stephen.Springer@nuance.com

of speech recognition, specifically the abilities and limitations of recognition of *novel* (first-time) callers in uncontrolled acoustic environments, such as cars, caf  s, and kitchens. While critical, the focus on only the technical challenges fails to consider the larger question of user satisfaction with, and adoption of, these interfaces.

Speech-activated interfaces for Call Centers, in particular, differ from other speech interfaces in some very important ways. For example, a speech interface on a computer, in a car, or on a mobile phone are all likely to have screens and physical controls, such as buttons or keyboards, that present distinct graphics and suggest specific modes of use. Moreover, graphics and physical controls provide a constant reminder that one is interacting with an engineered *thing*. By contrast, *speech-only* interfaces, as when one phones a Call Center, present only a voice on the other end of the phone. Earlier research [1, 2] has demonstrated that users will anthropomorphize a computer, especially when a speech interface is employed to control it. This effect of anthropomorphizing is magnified when one's only feedback is through a natural-sounding voice on the phone because in such instances there is no reminder that one is *not* interacting with a person. As such, we have seen time and again those users across the board will expect this "system" to respond as fluidly and universally as a person would. And why not? In fact, in order to encourage adherence to the rules of social etiquette (e.g., "it's polite to answer a question rather than ignore it"), designers of speech-only interfaces build systems that emulate human Customer Service Representatives (CSRs) as closely as possible. They do this because a conversation with a CSR, as opposed to a speech interface, is generally considered to be the caller's *preference*, but the expense of providing enough such CSRs in call centers can be prohibitive. An automated interface which emulates such capable human agents, then, may help the caller self-serve, without making it painfully obvious to the caller that they are getting something less than their ideal preference. And likewise, if the self-service can be completed without the need to transfer to a CSR, the company saves a considerable amount of money.

Of course, most callers are completely aware that they *are* dealing with an automated system. Despite this, it seems that most of us cannot help but react to an automated system's questions, vocal cues, and intonation as if we were speaking with a person. In one classic exchange, a caller exasperated with a speech-only interface in a Call Center exclaimed, "*Lady, if you can understand English, you can understand what I'm sayin' – I want to speak to a living body, please.*"

This curious paradox – knowing that one is speaking with a computer, yet speaking with it as if it were a live person – creates a unique challenge for speech-*only* interfaces. Whereas most users would not try to use a voice-activated GPS device on their dashboard to tune their radio, or a voice-activated mobile phone in their hand to turn up the lights, they have no reservation about asking a speech-only interface for anything that might reasonably be considered a part of Customer Service, just as they would do when speaking to a real human being. Consider the irony in that no one ever tries to "press 11" on a TouchTone™ system when only choices 1–4 are offered, but these same individuals without hesitation will ask a speech interface to do just about anything related to Customer Service. For this reason alone, it can be remarkably difficult to meet the caller's expectations for the speech-activated

self-service system operating in the Call Center. What is worse is that violating such expectations is a sure step toward the caller pressing zero, demanding an operator, or otherwise abandoning the self-service path. A successful speech system designer, on the other hand, will work hard to understand a caller’s expectations of technology, self-service, and Customer Service, and wherever possible, meet or beat those expectations. The most successful recipe for interaction will include these steps:

- Creating Caller Archetypes
- Addressing callers’ motivations to use the phone channel
- Reacting to issues, not blindly proffering solutions
- Understanding common reactions to everyday technology
- Bringing the transparency of real-world self-service systems to the phone
- Addressing the psychology of queueing theory

8.2 Creating Caller Archetypes

A technique that is common to many different design endeavors – but too often unused in Call Center application design – involves the creation of *Design Personas*, also known as *User Archetypes*, or, for purposes here, *Caller Archetypes*. An archetype is a concise, focused description of a specific yet fictional individual who may encounter the system. The archetype is given a name so that the designer and his reviewers can more easily envision the use of the proposed system by a real life user [3]. In the context of Call Center Self-Service, one can perhaps use 5–10 archetypes to capture a wide array of user expectations, behaviors, goals, and knowledge. Design elements are then evaluated against each archetype to ascertain if the suggested interaction is likely to encourage “expected” behavior.

For example, imagine two Caller Archetypes, Tim and Brigitte, for a prototypical credit card application. Tim is a 47-year-old salesperson with an excellent FICO score who pays his credit card statement balance in full each month by mailing in a check to the credit card company. Brigitte is a 28-year-old graphic artist who has access to the Internet most of the day at her workplace, runs a balance on her card, and pays online most of the time. When considering Tim’s payment history, a call arriving from him shortly after his payment was received, and no doubt right around his payment due date, might be quite effectively addressed by proactively offering him confirmation that his payment was received “on time” – before he has even requested such confirmation. In contrast, a call from Brigitte, made within minutes after she has paid part of her balance online, is much less likely to be about confirming her payment. Instead, the warrant for her call may be to negotiate a more favorable interest rate.

From this, we can see that the creation and use of effective Caller Archetypes constitute a requisite first step in the ongoing modeling of how a caller will react to a speech-only interface.

8.3 Addressing Callers' Motivations to Use the Phone Channel

Call Center professionals, looking to install a speech-only interface as a way to expand self-service over the phone, commonly turn to their Web site usage (and the backend functionality supporting it) in order to get a handle on services to offer over the phone. For example, it might be the case that our hypothetical credit card company has a Web site in which the two most popular destinations are (1) total balance, last statement balance, and due date and (2) Frequently Asked Questions. One might then reasonably assume the same self-service options that users exploit when using the enterprise's Web site will be similarly sought over the phone.

Such a supposition, however, fails to take into account that the user's choice to use the phone channel, as opposed to the internet, is a *deliberate* choice on the part of the caller. In fact, most of us do business quite happily with our bank, our utilities, our health care providers, and so on, without having to reach for the phone. Indeed, it's in these companies' best interests to anticipate our needs and create a *system of interaction* wherein phoning is unnecessary, since having a CSR field a single call is far more expensive than supporting a visit to a Web site, mailing a statement, or processing a cheque. Another advantage in using a company's Web site to conduct business is that the pages on the Web site are *persistent* – they stay on one's screen until one leaves the page, allowing for plenty of multitasking and casual browsing.

Against this mode of operation, by now familiar in everyday life, these days the customer's decision to phone a company is most often associated with the belief that something is "wrong" – the billed amount is wrong, the statement is missing, something is misordered, and so forth. When the customer surmises that something within the usual system of interaction is wrong, thereby necessitating a call to the enterprise, he will first need to reserve a few moments to reduce in-room distractions, plan an explanation (or argument) in order to get the system corrected, and then to concentrate on the phone call itself. This takes pretty deliberate planning. And should the customer go ahead and place the call, what will be his likely posture toward the speech-activated self-service system? That interface is after all an integral *part* of the overall enterprise system – a system that has obviously gone wrong, or else he would not have had to call in the first place. In such situations, one's natural inclination is to immediately begin seeking a way *around* this troublesome system, in order to reach a human being with free will – or at least approval from their manager – to correct the fault in the system.

This is not to say that it is entirely a mistake to provide the same services over the phone as are typically provided over the web or via the mail. A speech interface, however, that *only* offers a menu of a few prepackaged actions is likely to be viewed as an obstacle *between* the caller and his wished-for solution, rather than being seen as the vehicle for *effecting* the solution.

8.4 Reacting to Issues, Not Blindly Proffering Solutions

So, if a caller is to perceive an automated system as an ally, and not an obstacle, what is the best way to present that system? We can actually learn the answer by taking a walk to our neighborhood pharmacy.

Enter almost any pharmacy and observe how the aisles are labeled. You will see headings above the store aisles such as “Eye Care,” “Colds and Flu,” or “Aches and Pains.” Conversely, such store headings are almost never labeled, “Saline Solution,” “Dextromethorphan,” or “Antihistamines.” In other words, self-service is structured around the problematic *issues* that customers have, and not around the *solutions* available to them. What is interesting about this is that there are typically far more issues than solutions, at least in the self-service aisles. After all, there are probably hundreds of reasons why one might need an over-the-counter pain reliever, and yet the great majority of those are treated with one of the three common medications – aspirin, acetaminophen, or ibuprofen. Still, it is rightly perceived by the pharmacy that the customers need simply think of their problem, and it is then the pharmacy’s responsibility to suggest a solution, either by simply locating the solution under the problem heading above the aisle, or through the advice of their store pharmacist.

Compare this model of self-service with what commonly occurs in self-service phone systems. A call to the telephone company might be answered with, “Which would you like: billing, orders, or repair?” It is a straightforward question, likely designating three main departments in the corporation. But notice that it begins by asking the customer to pick a *solution*. It may not seem like it, but such a setup places a mental strain on the caller, who must draw a line from what he was thinking of as his problem (e.g., “It looks like I’m being charged for something that isn’t activated yet.”) to the department that is most likely to provide the solution. Often, such mapping between problems and solutions is not clear. The result? Many callers will forego self-service, and seek the assistance of the CSR.

This is one of the great values in the growing trend to move speech-activated self-service from directed-dialog menus (“Please choose billing, orders, or repair”) to Semantic Language Models (SLMs), which instead ask “How may I help you?” In fact, such open-ended systems might ultimately just route the caller to one of the three departments. The difference is that it is the *company*, and not the caller, that takes on that mapping task. The caller is simply encouraged to say what was on his mind. Given the rules of social etiquette, which are accordingly much more in play with interfaces that one *speaks* with, it is actually harder for callers to “refuse to play” by not answering this most obvious of questions. And once they *have* said how they might be helped, the self-service conversation has at least begun.

8.5 Understanding Common Reactions to Everyday Technology

A caller’s willingness to engage with self-service is a start. But it is only a start. The successful system designer must constantly reinforce that it was the *right* start that the caller has not made a mistake by beginning a “conversation” with an automated technology.

This is trickier than it may seem. Designers of Interactive Voice Response (IVR) systems know full well that callers are predisposed (presumably from prior experience) to be inimical to IVR systems. In fact, it is probably more correct to say that consumers can be biased against *all* technology, except that which is authentically,

surprisingly simple. We all find DVD players, microwaves, even electronic thermostats extremely useful parts of our everyday lives. But when first encountered, do not they *all* cause some amount of consternation and angst? We muddle through on the right set of keys to press in order to reheat a cup of coffee, and then often cling to that knowledge as our main understanding of how to get *past* a confusing interface to a desired solution.

The general rule for making technology approachable – obvious as it sounds – is to make its interface *extremely* simple and intuitive. The Apple iPod and the Nintendo Wii were both almost instantly iconic not just for their functions, but for how obviously simple their designs were found to be. Compared with their competitors, who often layer one feature on top of another, one could almost *see* success in their future use.

This rule is often forgotten in the design of Call Center Self-Service, where, in the name of incrementally increasing automation, options upon options are added over time. It is the self-service equivalent of building a microwave with “one-touch reheating of Tuna Casserole” – yes, one-touch for the very few who bother to memorize the 133 touches available to them, but simply confounding to the rest of us. The successful designer of a speech-only interface would do well to remember that the instant a prospective user senses that a technical interface will be the least bit complicated, he will distrust it, expect failure, and look for escape routes. Given the ubiquity of self-service IVRs with options for “more options,” callers have already had their expectations set that a new IVR will be as impenetrable as the interface to a new microwave. Yet, we can beat these expectations by deliberately looking for all possible ways to simplify every single choice and every single path in the IVR. Speech interfaces already allow us to ask questions naturally and to understand callers’ natural language responses. That is, SLMs (e.g., “How may I help you?”) help us to avoid exhaustive enumeration of options from which to choose. Consider that microwave designers seem determined to add obscure features such as “one-touch reheating of Tuna Casserole,” which theoretically meets the goal of 0.5% of users – despite the likelihood that too many of these options bewilder the remaining 99.5% of users. Similarly, adding option upon option to the speech-only interface may help a very limited subpopulation of callers, but has a distinctly negative, confusing effect on everyone else.

8.6 Bringing the Transparency of Real-World Self-Service Systems to the Phone

As we can learn from pharmacies and microwaves, so, too, can we learn from about the value of *transparency* from other real-world self-service systems. At the airport, for example, a check-in line might have “self-service check-in” kiosks. Three subtle features of their placement are worth noting:

- They are always placed directly next to the “wait in line for an agent” queue – anyone choosing one over the other is free to switch lines at any time.
- The approaching customer can spy in an instant how many people are waiting in line, and how quickly the line is moving.
- Seasoned travelers are likely as well to size up the particular people in line: those with extra luggage, small children, large parties, or simply confused expressions might reasonably be expected to make for a longer wait than a queue of serious-looking business travelers with only carry-on luggage.

Consider how this relatively simple setup at the airport self-service check-in differs from the average set of options provided in most telephone self-service systems. We are asked to “press 1 to access our automated system,” a commitment that is hard to reverse – or at least, perceived to be. We are asked to choose up front between self-service and live help, almost always without any ability whatsoever to divine the wait time for live help. We have no visibility into who else is waiting, or even that there are actual *people* ahead of us – instead, if we are told about the wait at all, it is in completely disembodied terms of “the expected wait time is approximately three minutes.” That is an interesting formulation, in that it removes from view *why* we have to wait in the first place! All of these characteristics conspire to remove *transparency* from our choice. And in the real world, whenever anyone asks you to make a choice without giving you the basic information to do so, you are likely to distrust that person and hasten to seek an escape.

Consider, however, how the following hypothetical interaction might actually *exceed* the caller’s expectations:

- System: Hi, thanks for calling Acme Airlines! Let me find a representative who can help you. <brief music> Okay, it looks like the agent with the shortest line still has three people ahead of you. Hold on just a moment.
- System: <more music>
- System: By the way, if you’re calling to confirm a reservation or to check in, just say “self-service” at any time – I’ll hold your place in line in case it doesn’t work out. Hmm, still three people in line....
- Caller: Self-service.
- System: Sure! If you want to grab your place back in line anytime, just press zero.
- System: <chime> (new voice) Hello. Do you have your confirmation number handy?
- Caller: Yes, it’s, uh, S, R, S, 1, 2, 3...

Such a system begins to approximate the transparency and ongoing choice one has in real-world self-service situations. By not shying away from presenting options that most callers naturally assume are hidden there anyway (i.e., live CSRs in a call center) we can remove the sense of “pushing” the caller into something he may not like – and at the same time garner enough good will to automate more calls than by forcibly herding every single caller through these self-service systems.

8.7 Addressing the Psychology of Queueing Theory

The hypothetical example above reveals one other aspect of real-world Customer Service that shapes a caller's expectations: the psychology of queueing theory that is often employed in lines at fast food restaurants.

There are two competing models. In one, customers are directed into a single snaking queue, a “first-in, first out” model, whereby multiple servers call upon the next person in line. In the other, customers are invited to choose the server they want to wait for, from a set of several independent lines, and must wait for a particular server. In the second model, the lines are much shorter than in the first, but they do not move so fast, and any given line may be held up by a customer having trouble completing their order. The impulsive customers in a food court may be attracted to the shorter lines, but they are unlikely to get their food any faster.¹

What is interesting about queueing theory is not so much which kind of queue is better (the mathematics of this is well understood), but the *psychology* of the wait. We patiently wait for many things in the real-world, from a Web page loading to entry through a highway toll booth. The Walt Disney Company, at its theme parks, has poured enormous resources and energy into making the wait itself entertaining, and to great effect. In contrast, the psychology of the wait for *callers* still seems to be underappreciated as a fruitful area of study. In fact, the most common model of waiting “on hold” “for the next available agent,” with no clear indicator as the time remaining, has virtually no analog in the real world. It is the equivalent of asking a live customer to sit with his hands folded in a dark room until he is called, with no feedback provided as to how much time is passing, or how much time is left. And so virtually any wait is considered by callers and corporations alike to be a condemnation of the caller’s significance. Since no company wants to be perceived as “not caring about their customers enough to staff their call centers sufficiently” (despite the fact that so many real-world waits for service are longer than the typical 2-min hold time), call centers are heavily staffed, and wait time is consequently minimized to the point that there is often little “upside” to choosing the self-service route instead of a human agent. We believe that there are, in fact, many different options for making a longer queue time *feel* more tolerable. If these options were to be employed, this would in turn allow for actually longer wait times, which could be used strategically to *promote* self-service selection, as an alternative to the all-too-familiar wait time for a CSR.

8.8 Summary: Meeting and Beating Expectations

Speech-only interfaces and accessed over a phone are often conceived as a union of three separate elements: a speech recognizer detecting input; a set of call-flow logic describing functionality; and a set of prompts designed to elicit the “right” reaction

¹In fact, given the freedom to switch lines, a customer will, on average, wait the same amount of time in either queue formation – though the multi-queue customer will experience a greater variation in wait times from one meal to the next.

from callers. But users of such a phone interface are hardly puppets, to be manipulated by clever tweakings of prompt-wordings. Instead, users arrive at a speech-only interface with specific ideas of what they need to accomplish, with fairly negative *a priori* attitudes toward technology in general, and with a benchmark for comparison drawn from plenty of real-world “self-service” experiences, which all too often are not reflected in a call center’s self-service platforms.

To meet, and perhaps sometimes even beat, these callers’ expectations, the designer of a speech-only interface must attend to several key considerations:

- The careful and creative use of Caller Archetypes can ensure that most design decisions are evaluated with respect to the experiences and expectations of the various specific callers to a particular application, and not in response to the system designer’s, or his client’s, own predispositions.
- The phone service must present itself not just as another automaton relocated to a new medium, but as an active participant and aide in a caller’s attempts to fix problems and get questions answered.
- The application should allow callers to easily state their issue or concern and then present a selected solution to them, instead of asking callers to select their own solution from a menu of perhaps indeterminate “options.”
- Incremental additions of functionality can introduce more complexity than they address, and should therefore be added very carefully.
- Transparency is important to customers. Enormous opportunities may exist in recasting phone systems more in the vein of “real-world” self-service opportunities, which often make advantages and disadvantages of competing solutions visible, and can provide a compelling case for the customer to deliberately choose self-service over live help.
- The psychology of queueing theory can be an important ally in building a positive user experience. The act of waiting can be artfully transformed into a far more pleasant experience than it is on its face, and the potential for avoiding an (even pleasant) wait can be exploited to encourage the caller to seek a positive self-service experience.

Designing a compelling conversation with a speech-only interface presents unique challenges. We can improve these conversations not just by understanding the technology, and not just by word-smithing prompts, but by actively embracing many of the real-world experiences and expectations of callers as we design these systems.

References

1. Reeves B, Nass C (1998) *The media equation: how people treat computers, television, and new media like real people and places*. Cambridge University Press
2. Nass C, Brave S (2005) *Wired for speech: How voice activates and advances the human-computer relationship*. The MIT Press
3. Cooper A (2004) *The discussion of user personas and scenarios. The inmates are running the asylum: why high tech products drive us crazy and how to restore the sanity*. Sams

Chapter 9

“For Heaven’s Sake, Gimme a Live Person!” Designing Emotion-Detection Customer Care Voice Applications in Automated Call Centers

Alexander Schmitt, Roberto Pieraccini, and Tim Polzehl

Abstract With increasing complexity of automated telephone-based applications, we require new means to detect problems occurring in the dialog between system and user in order to support task completion. Anger and frustration are important symptoms indicating that task completion and user satisfaction may be endangered. This chapter describes extensively a variety of aspects that are relevant for performing anger detection in interactive voice response (IVR) systems and describes an anger detection system that takes into account several knowledge sources to robustly detect angry user turns. We consider acoustic, linguistic, and interaction parameter-based information that can be collected and exploited for anger detection. Further, we introduce a subcomponent that is able to estimate the emotional state of the caller based on the caller’s previous emotional state. Based on a corpus of 1,911 calls from an IVR system, we demonstrate the various aspects of angry and frustrated callers.

Keywords Interactive voice response system (IVR) • Call center • Anger detection
• Angry user turns • Emotional state of caller • Angry or frustrated callers • Dialog manager • Natural language • Support vector machine (SVM) • Discourse features
• Acoustic modeling

9.1 Introduction

More and more companies are aiming to reduce the costs of customer service and support via automation. Recently, with respect to telephone applications, we have witnessed a growing utilization of spoken dialog technology in the call center [4]. Such interactive voice response (IVR) systems, called as such for they allow for an

A. Schmitt (✉)

Scientific Researcher, Institute for Information Technology at Ulm University,
Albert-Einstein-Allee 43, 89081 Ulm, Germany
e-mail: alexander.schmitt@uni-ulm.de

interactive control of a telephone application via voice or touch tone, are being used across various domains.

The first generation of IVRs was entirely touch-tone based. The caller could navigate through an auditorily presented menu by using the keypad on the telephone. This is known as dual tone multiple frequency (DTMF). Among the first applications were call routers, which served as front-ends that helped to find the suitable contact person in a company. Apart from performing this function, such applications had mostly an information retrieval character: the user provided a piece of input; the system replied with a piece of information [17].

With the progress in automatic speech recognition (ASR), systems increasingly came up with the ability to be controlled by voice instead of DTMF. Some systems combined DTMF with voice, making ASR technology complementary to DTMF. There is no doubt that voice enabled richer IVR applications. But at the same time, voice introduced a new vulnerability to those applications given the fact that speech recognition errors were likely to occur.

Systems of this generation offered predefined choices to the user, such as a navigation setup that is based on command-style speech input. Similarly, information retrieval and, later, transactional applications made good use of this technology. For example, companies launched hotlines enabling customers to track packages, retrieve information about trains schedules, book hotels or flights, perform stock trading, or manage bank accounts.

Yet, the advances in speech technology allowed for another generation of IVR applications entailed with problem-solving capabilities [1]. Such applications have found their niche in automated technical support. The most recent generation of IVRs make it possible for the caller to describe the reason of the call using natural language (NL). Instead of command-based input, which makes it difficult for the caller to select among a huge list the possible reason(s) for his call, NL affords the user the ability to describe his problem or concerns in complete sentences even though the subsequent dialog is mostly still command-based. Not surprisingly, the complexity of those systems rose substantially. Just imagine, while early IVRs consisted of only few dialog steps, today's problem-solving applications may contain several dozen, and frequently up to 50–100 dialog steps just in one call.

Notwithstanding this complexity, both customers and providers still have a substantial interest that the call in which they invested a substantial amount of time ends up successfully. Both in fact have their own pressures: While the customer is at the risk of futilely spending time with an IVR system that might not solve the problem which necessitated his call in the first place, the provider in return has running costs for each call that occupies a port on his telephone platform, not to mention loss of company image when the system dissatisfies the caller. The worst case scenario is a situation in which callers having spent a substantial amount of time with the system are forced to hang up.

These changing conditions make it more necessary than ever to permanently monitor the ongoing conversation and to instantly offer a solution to problematic situations before the caller decides to hang up.

Potential solutions could be as follows:

- The dialog manager launches a subroutine, trying to repair the current situation or switches to a more restricted dialog strategy with more explicit confirmations
- The system automatically escalates to a human operator, or
- Human operators permanently monitor the number of ongoing calls on a status screen, deciding to step in when the system detects problematic dialog situations that portend that the call is about to fail

In this chapter, we consider calls where users are getting frustrated or angry to be “problematic.” There are some approaches that tie the term “problematic” to task completion and we will quickly sketch some of this work in Sect. 9.3, where we will discuss related work. However, this chapter is all about anger and we illuminate the various aspects of detecting frustrated callers. Although acoustic information plays a predominant role, we also consider other information sources.

The chapter commences with a real-life example of a situation, where a caller becomes frustrated, and continue with a description of related work in the field of detecting “problematic” dialogs and anger recognition. Pivotal to this chapter is the section that provides a description on developing a speech corpus that serves as training and test material for an anger detection system.

After that, in the next section we discuss how to detect angry and dissatisfied callers based on acoustic information. Acoustic features are presented and their individual impact on the task of detecting anger is shown. To what extent linguistic information can be used in a practical sense for predicting frustration is discussed and demonstrated in the following section. In a third step, we demonstrate that it is even possible to a certain extent to identify angry dialog turns, even without any acoustic information, relying instead on dialog and contextual features. In the penultimate section, we demonstrate that angry user turns rarely occur in isolation and thus exploit this information for building a subsystem that predicts anger based on previous frustration. Finally, all described approaches are combined in the last section, where we present an anger detection system along with an evaluation of its overall performance.

9.2 An Exemplary Dialog

Before we delve into the various aspects of anger detection, we first take a closer look at an example of a dialog situation, where the caller gets frustrated and angry in a deadlocked situation. The data and the examples throughout this chapter originate from a technical support automated agent that is able to resolve Internet-related problems jointly with the caller. The system is able to regain connectivity, deliver lost passwords, or schedule appointments with service employees. Table 9.1 presents a critical situation: in order to be able to proceed, the system requires the brand name of the caller’s modem.

Table 9.1 Example dialog of a typical emotional situation caused by misunderstandings: The system asks the caller to identify her modem and tell the brand name. The caller is pretending that there is no modem, a case the system does not intercept since the nonexistence of a modem would not allow for an internet connection at all. Both the system and the user are overstrained by this situation and the caller is getting angry. The caller's answers comprise hand transcriptions, natural language parse, and emotional state

☺= non angry; ☹= annoyed; ☹☹= angry

| | |
|--------|---|
| Agent | So you have a Netgear modem, is that right? |
| Caller | yes [yes] ☺ |
| Agent | Before we go on, I need you to make sure that Netgear has a light on the front labeled "Status." When you've found out whether or not your Netgear has a Status light, say Continue |
| Caller | continue [continue] ☺ |
| Agent | [...] So, is there a Status light on the modem? |
| Caller | no [no] ☹ |
| Agent | Well, if it doesn't have that Status light, then I think you're looking at your Netgear *router*. [...] When you've found your modem, say Continue |
| Caller | noise [nomatch] ☹ |
| Agent | Please, just say Continue after you've found your modem |
| Caller | continue [continue] ☺ |
| Agent | What's the brand name? |
| Caller | noise [nomatch] ☹ |
| Agent | I didn't quite get that. Please tell me the brand of your modem again, or, if there is no brand printed on it, say No Brand Name |
| Caller | no modem [no brandname] ☹ |
| Agent | So your modem does *not* have a brand name printed on it, right? |
| Caller | there is no modem [nomatch] ☹ |
| Agent | So your modem does *not* have a brand name printed on it, right? Please say Yes or No |
| Caller | no modem [nomatch] ☹ |
| Agent | So your modem does *not* have a brand name printed on it, right? Please say Yes or No |
| Caller | no [nomatch] ☹ |
| Agent | Let's just move on for now |
| Agent | Now that we've correctly identified the modem, I need you to unplug it. When it's unplugged, say Continue |
| Caller | no, there is no * modem! give me a * person! [nomatch] ☹ |

While the caller insists that he has no modem, the system insists in asking for the modem's brand. Certainly, the dialog manager is working correctly at this moment since the caller must have a modem in order to go online. However, situations like these could potentially be prevented if the dialog system would be able to detect that the caller is becoming frustrated. This could either happen in an online manner (real time) in which the system permanently monitors the emotional state of the caller and escalates to a human operator once such a problematic situation is detected, or it can be done offline on logged and recorded dialogs to spot dialog steps where callers frequently get angry. One can see how anger detection could

also be of great benefit when system developers try to find out flaws in the dialog design for the purpose of improving the overall system.

The reasons why callers become frustrated or angry are various: subsequent misrecognitions due to background noise; out-of-scope prompts from the user side; inappropriate grammars; or simply a poorly working automated speech recognizer. In some instances when callers do not expect to have their inquiries handled by automation, they can become annoyed right away when the call is answered by an automated system. Sometimes, users are known to try to bypass the system by shouting at it, assuming that there is an emotion recognizer deployed. Of course, in most cases, this is not (yet) the case.

9.3 Related Work

Detecting problematic dialog situations in customer self-service is not necessarily an acoustic, and thus, an anger detection task.

Some of the first models to predict problematic dialogs in IVR systems were proposed by Walker et al. [13, 25]. They employ RIPPER, a rule-learning algorithm, to implement a Problematic Dialogue Predictor forecasting the call-outcome of calls in the HMIHY (How May I Help You) call routing system from AT&T [9]. The classifier is able to determine whether a call belongs to the class “problematic” or “not problematic” and employs the classifier’s decision to escalate to a human operator. “Problematic” in this context are calls whose task completion is endangered. Due to the nature of HMIHY, the dialogs are quite short with not more than five dialog turns. Walker et al., respectively, built classification models based on features extracted out of the first dialog exchange, and another model based on features from the first and the second exchange. By virtue of that, the system is able to detect a problem directly after having seen the first exchange by using the first model with an accuracy rate of 69.6% and with 80.3% accuracy after having seen two exchanges. And because of this the decision point is fixed.

Walker et al. inspired further studies on predicting problematic dialog situations:

- van den Bosch et al. [24] report about online detection of communication problems on the turn level by using RIPPER as classifier and the word hypothesis graph plus the last six question types as training material. If communication problems, i.e., misrecognitions, are detected, the authors propose to switch to a more constrained dialog strategy. Since users tend to speak intentionally loud and slow when facing recognition errors – a situation in which a conventional speech recognizer has no training – these authors propose the use of two speech recognizers in parallel in order to detect hyperarticulated speech more robustly. Note that the aim is not escalation but adaptation.
- Levin and Pieraccini [15] combined a classifier with various business models to arrive at a decision to escalate a caller depending upon expected cost savings in so doing. The target application is that of a technical support automated agent. Again a RIPPER-like rule-learner has been used.

- In [21], we presented an approach similar to [15] that demonstrates expected cost savings when using a problematic dialog predictor for a technical support automated agent in the television and video domain. Under the hypothesis that acoustic features extracted from caller utterances support the detection of problematic situations, we carried out a study that incorporated average pitch, loudness, and intensity features within each dialog exchange [10]. A visible impact, however, could not be observed.
- Paek and Horvitz [16] considered the influence of an agent queue model on the call outcome and included the availability of human operators in their decision process.
- A rather simple, yet quite effective approach has been published by Kim [11], where a problematic/non-problematic classifier that is trained with 5 g of utterances from callers¹ reaches an accuracy of 83% after five turns. Escalation is performed when the quality falls below a certain threshold.
- Zweig et al. [26] present an automated call quality monitoring system that assigns quality scores to recorded calls based on speech recognition. However, the system is restricted to human–human conversation and the aim is to survey whether operators behave courteously and appropriately in dealing with customers.

An increasing number of studies analyze speech-based emotion recognition and anger detection in telephone-based speech applications.

Offering as much as 97% accuracy for recognition of angry utterances in a seven class recognition test performed by humans, the TU Berlin EMO-DB [5] bases on speech produced by German-speaking professional actors. Here it is important to mention that the database contains ten preselected sentences all of which are conditioned to be interpretable in six different emotions and neutral speech. All recordings have wideband quality. When classifying for all emotions and neutral speech automatically Schuller [23] resulted in 92% accuracy. For this experiment he chose only a subset of the EMO-DB speech data that, judged by humans, exceeded a recognition rate of 80% and a naturalness evaluation value of 60%. Eventually, 12% of all utterances selected contained angry speech. He implemented a high number of acoustic audio descriptors such as intensity, pitch, formants, Mel-frequency Cepstral Coefficients (MFCCs), harmonics to noise ratio (HNR), and further information on duration and spectral slope. He compared different classification algorithms and obtained best scores using support vector machines (SVM).

A further anger detection experiment was carried out on the DES database which contains mostly read Dutch speech and also includes free text passages [8]. All recordings are of wideband quality as well. The main difference to the EMO-DB is that the linguistic content had not been controlled entirely during recordings. The people chose their words according to individual topics. The accuracy for human anger detection for this corpus resulted in 75%. This accuracy is based on a five class recognition test. Schuller results in 81% accuracy when classifying for all emotions. Voting for maximum prior probability class would reach an accuracy of 31% only.

¹A sequence of five consecutive user turns.

Note that these studies and the results are based on acted speech data, containing consciously produced emotions, performed by professional speakers.

Lee and Narayanan [14], as well as Batliner [2] used realistic IVR speech data. These experiments use call center data, which is of narrow-band quality. Also the classification tasks were facilitated. Both applied binary classification, i.e., Batliner discriminates angry from neutral speech, Lee and Narayanan classify for negative vs. non-negative utterances. Given a two-class task, it is even more important to know the prior probability of class distribution. Batliner reaches an overall accuracy of 69% using linear discriminative classification (LDC).

Lee and Narayanan reached a gender-dependent accuracy of 82% for female and 88% for male speakers.

9.4 Getting Started: The Basic Steps Toward an Anger Detection System

The core of an anger detection system is a model that contains the characteristics of angry user utterances and non-angry user utterances. In other words, the model allows us to classify and “detect” the emotional state of an user where the emotional state is unknown. In order to obtain such a model, four basic steps are required:

Data Collection: First, exemplary user utterances have to be captured that serve as training material. In this context, each user utterance is also called sample or example. An anger detection system works best when the training material originates from the same system and has been captured under the same conditions as in the data that is later classified within the live system.

Labeling: Second, since the emotional state of the caller in each specific turn is unknown and we are dealing with non-acted emotions, a label has to be assigned to each sample in a manual rating process. Best practice is to take into account the opinion of several raters listening to the samples and to assign the final label based on majority voting.

Feature Extraction: Third, features that indicate anger are extracted from the captured data. Particularly acoustic information will be used. Later, we demonstrate that non-acoustic information as well, which has been logged during usage of the dialog system, can also be of benefit for determining the emotional state of the caller.

Training: Fourth, we engage in the task of classification which is to map a prediction on an unknown sample based on the features we extract from the unknown sample. To achieve this, we apply a supervised machine learning algorithm. It is called “supervised” since we present both the algorithm feature-set of a sample plus the label of the sample. There are a variety of supervised learning techniques. Although the most well known among such learning techniques is artificial neural networks (ANN), at the same time other techniques such as Nearest Neighbor, Rule Learner or SVMs, are frequently used. Depending on the task and the data, a certain technique might be known to perform better than another.

9.5 Corpus

All tasks in machine learning require training material, no matter if we are developing a system that is able to visually detect traffic signs, to recognize speech and handwriting, or, as in the present case, frustrated callers. For our study, we employed 1,911 calls from the automated Internet agent containing roughly 22,000 utterances. Since we are dealing with non-acted data and thus are not able to ask the caller about her emotional state, we have to estimate the emotional state on our own. In our scenario, we asked three labelers to listen to all utterances and assign a label for each sample. Since it seemed to be crucial to determine whether the caller was non-angry or angry, we introduced additionally to the labels “angry” and “non-angry” a third label that we call “annoyed” to ease the rater’s decision in the case of doubt or when he felt that the caller is only slightly angry. For non-speech events or cross-talk, the raters could also designate the label “garbage.” Finally, the corpus could be divided into “angry,” “annoyed,” “non-angry” and “garbage” utterances. The final label was defined based on majority voting, i.e., when at least two of the three raters voted for “angry,” the final label that was assigned to the sample was “angry.” The final distribution resulted in 90.2% non-angry, 5.1% garbage, 3.4% annoyed and 0.7% angry utterances. For 0.6% of the samples, all three raters had different opinions and no agreement on a final label could be achieved. While the number of angry and annoyed utterances seems very low, 429 calls (i.e., 22.4% of all dialogs) contained annoyed or angry utterances. For details on the ratings see Fig. 9.1.

For training the subsystems that we present in the course of this chapter, we employed a subset of the data and removed a large amount of non-angry utterances. We collapsed annoyed and angry utterances into one class that we call angry and created a test and training set according to a 40/60 split in order to prevent a bias

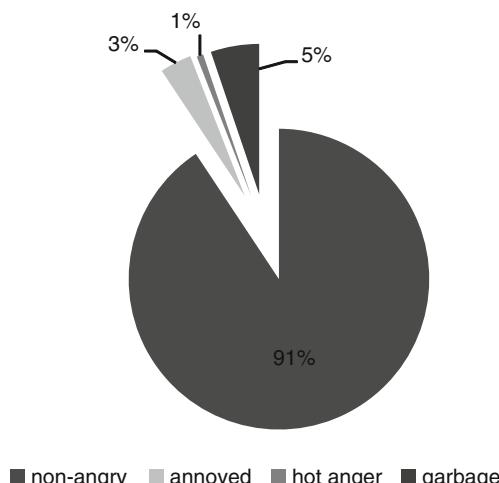


Fig. 9.1 Distribution of final labels after rating process on complete corpus

toward the non-angry class. The resulting sub set consists of 1,396 non-angry and 931 angry turns. Note that the system is speaker independent since speakers that were used for training the system did not occur in the test set.

9.5.1 Inter-rater Agreement

To measure the degree of agreement between raters, the Cohen’s Kappa coefficient [7] expressing the inter-rater reliability is frequently used. Cohen’s Kappa takes into account that agreement between raters might also happen by chance and creates a more reliable statement on the agreement than a simple percentage calculation would do. The agreement between two raters is calculated as

$$k = \frac{p_0 - p_c}{1 - p_c},$$

where p_0 is the relative agreement between the raters and p_c is the hypothetical agreement by chance.

To generate a robust classifier, a clear separation of the patterns, in our case angry and non-angry user utterances, is mandatory. A too low κ would potentially lead to a non-robust classifier in the final system.

To put it simply: How could a machine-learning algorithm be able to separate patterns that even humans have difficulties in doing?

The agreement in our final subset on the three different classes by all three raters resulted in $\kappa = 0.63$, which can be interpreted as substantial agreement [12]. Details of the corpus are listed in Table 9.2.

Table 9.2 Details of the Internet agent speech database

| Domain | Internet support |
|---|---------------------|
| Number of dialogs in total | 1,911 |
| Duration in total | 10 h |
| Average number of turns per dialog | 11.88 |
| Number of raters | 3 |
| Speech quality | Narrow band |
| <i>Deployed subsets for anger recognition</i> | |
| Number of anger turns in trainset | 931 |
| Number of non-anger turns in trainset | 1,396 |
| Average duration anger in seconds | 1.87 s |
| Average duration non-anger in seconds | 1.57 s |
| Cohen’s extended Kappa | 0.63 |
| Average pitch mean anger | 205.3 ± 60.5 Hz |
| Average pitch mean non anger | 181.5 ± 63.7 Hz |
| Average intensity mean anger | 70.5 ± 6.3 dB |
| Average intensity mean non anger | 62.4 ± 6.1 dB |
| Average duration anger | 1.86 ± 0.61 s |
| Average duration non anger | 1.57 ± 0.66 s |

9.6 An Anger Detection System and Its Subsystems

Certainly, the distinction of angry callers is frequently an acoustic and, to a certain extent, a linguistic task. Additional information sources, such as video material or bio-sensors that are frequently used in emotion-detection research, are of course not available. On the other hand, information sources other than acoustic and linguistic sources can be exploited so as to indicate that a caller might be “angry.” Our system consists of four different subsystems, each of which estimates the emotional state of the caller and contributes to the final decision on whether the caller is currently angry or non-angry. Note that the system predicts turn-wise estimations, i.e., it is trained to detect the emotional state of the caller based on information from a single utterance. The complete system with its subcomponents is depicted in Fig. 9.2.

The first, most computationally-intensive subsystem is the acoustic subunit. It derives acoustic and prosodic features from the user utterance and detects frustration based on auditory events.

The second unit, a linguistic subsystem, spots anger based on words contained in the user utterance. Words being closely related to frustration, and frequently used in the specific domain to express anger, are the central information source.

The third subsystem is a dialog and a contextual subunit that exploits interaction parameters that are logged during dialog system usage, and also models the quality of the ongoing call. Our assumption is that subsequent misrecognitions of the ASR and frequent barge-ins² from the user are leading to or are an indicator for anger.

The fourth subunit models the previous emotional states of the caller and, by virtue of that, accounts for the fact that anger is rarely confined to one single turn; instead anger can be found in several subsequent dialog steps. Users do not get angry out of the blue. On most occasions, short of sudden sparks of anger that may happen precipitously, a certain history of anger – anger that builds up – can be observed when looking at calls containing angry user turns.

All four subsystems are later combined to a final classifier that allows a robust detection of angry user turns.

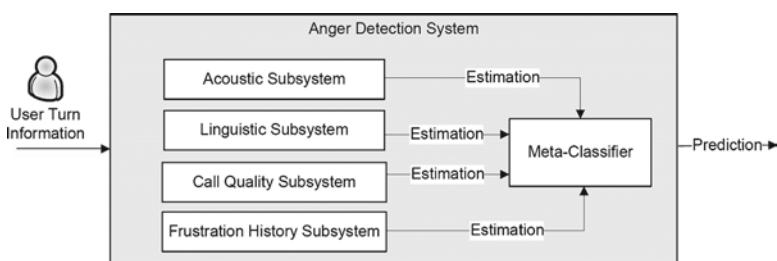


Fig. 9.2 Anger detection system consisting of acoustic, linguistic, call quality, and frustration history subsystem

²User interrupts the system prompt by speaking.

9.7 Acoustic Subsystem

Acoustics provide one of the most relevant sources of information to determine the caller’s emotional state. An acoustic and spectral analysis extracts relevant features that could indicate the emotional state of the caller. Note that, although we are talking of the emotional state, we restrict our system to detect “angry” vs. “non-angry”³ utterances.

9.7.1 Acoustic and Prosodic Features

Frustration is, at least in the context of telephone applications, first and foremost an acoustic sensation. If we take a closer look at the spectrograms of two utterances from a caller talking to the Internet troubleshooter, we can clearly see spectral differences (cf. Fig. 9.3). In both cases the caller says “steady on” to indicate that the LED on her modem is on. The first time, as shown in Fig. 9.3a, the caller speaks normally. After a misrecognition on the part of the dialog system, the user gets frustrated and shouts at the dialog system which is depicted in Fig. 9.3b.

The second utterance contains more energy which can be seen at the higher amount of yellow and red areas in the spectrogram. Especially in higher frequency bands we observe more energy than in the first utterance and it is clearly visible that the caller raised her voice. The second utterance contains a pause at about 0.3 s between “steady” and “on.” Presumably, the caller expects to facilitate recognition by isolating each word. By drawing upon these differences (such as separating words)

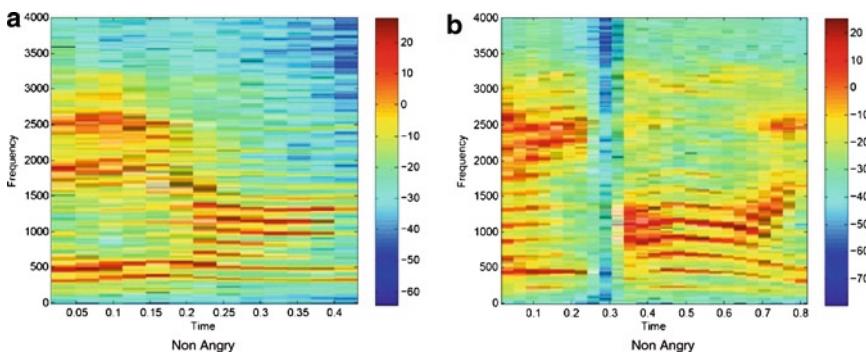


Fig. 9.3 Spectrogram of a caller saying “steady on” in a non-angry manner (a) and in angry manner (b) after being misunderstood by the system. Note that the angry utterance contains more energy in higher frequencies

³The term non-angry is used as an all-encompassing term for all emotions other than anger. However, since callers typically do not talk in a “happy,” “sad,” or “disgusted” manner to an IVR, “non-angry” speech contains predominantly neutral speech.

in combination with the spectral differences as depicted here, we can classify and distinguish between angry and non-angry user turns.

Our current acoustic subsystem [18] consists of a prosodic and an acoustic feature definition unit calculating a broad variety of information about vocal expression patterns such as pitch, loudness, intensity, MFCC, formants, harmonics-to-noise ratio, etc. Initially, the values depict average values that are calculated on the complete utterance. A statistical unit derives means, moments of first to fourth order, extremes and ranges from the respective contours. Special statistics are then applied to certain descriptors. Pitch, loudness, and intensity are further processed by a discrete cosine transform (DCT) in order to model its spectra. In order to exploit the temporal behavior at a certain point in time, we additionally append first and second-order derivatives to the pitch, loudness, and intensity contours and calculate statistics on them alike. The complete feature space comprises 1,450 features per user utterance.

9.7.2 Classification

Now that we obtained the features from the user utterances, the aim is to build a classifier that is able to determine the emotional state of an unknown user utterance. A fast and high performing classifier, and thus the classifier of our choice, will be a SVM. An excellent introduction to SVMs is provided in Bennett and Campbell [3]. SVMs are, although the name suggests it, not real machines. The term “machine,” however, stems from the fact that SVMs are machine-learning algorithms that use so-called Support Vectors. In simple words, a classifier is able to determine from an unknown sample (in our case an angry or non-angry user utterance) to which class it belongs by using a model that is based on a number of training examples. An example of such a model that has been derived from an SVM algorithm is provided in Fig. 9.4.

Special to SVMs in comparison to other classifiers is that they use hyperplanes to separate the training samples into two areas. An SVM considers training examples as points in an n -dimensional vector space. Instead of two dimensions as depicted here, our data points in the SVM will have various dimensions up to a maximum of 1,450. The hyperplane that is fit in between the two classes, angry and non-angry, will create a maximum margin between the two classes and is described by a set of original data vectors, which are therefore called Support Vectors.

9.7.3 Removing Redundancy

In a classifier intended for use in a deployed system, computational costs play a crucial role. However, calculating acoustic and prosodic features out of the speech signal that are not improving the classifier’s performance would be a waste of computational power. Moreover, calculating such features could harm recognition results insomuch as adding irrelevant information can confuse the choice of support vectors in the SVM. In a second step, we therefore rank all 1,450 features according

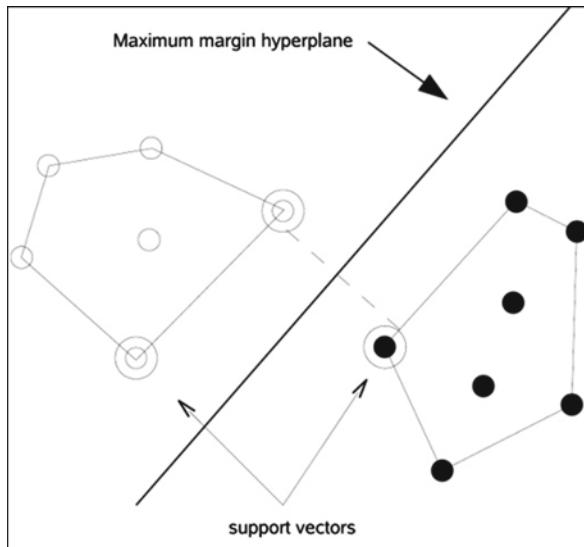


Fig. 9.4 A support vector machine (SVM) with a maximum margin hyperplane separating two classes, e.g., angry and non-angry user utterances

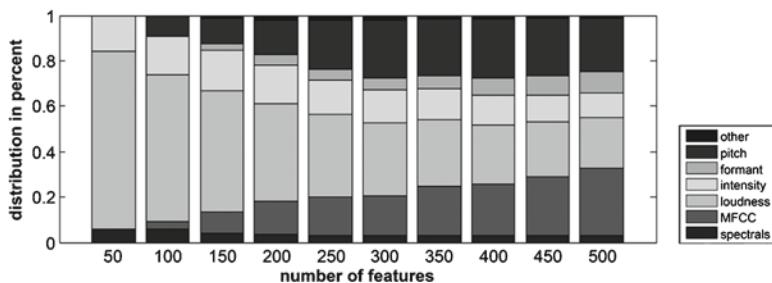


Fig. 9.5 Most relevant features according to IGR when considering the 50, 100, 150, etc. most relevant features

to their benefit to the classification task. A conventional method of determining the most relevant features in a set is the information gain ratio (IGR) ranking. We do not go into detail at this point, but refer to [18] for a more detailed explanation of IGR.

All samples consisting of the features and the label are subject to IGR. By virtue of that, we obtain a list of the average merit of each single feature for the classification task. Exactly how relevant features from a specific category are, in other words their scale of relevance, is depicted in Fig. 9.5. Loudness- and intensity-related features play the most relevant role since they dominate the group of the 50 most relevant features. Pitch, formants, and MFCC-related features play a subordinate role: they first appear when the 100 or 150 most relevant features are considered.

One might assume that the more features we use for our classification, the better the performance of the classifier. Too many features, and especially those that are

irrelevant, might harm the performance of the final system. In an iterative process, we therefore determine the optimum number of features for our SVM. Beginning with the top-most feature according to our ranking list, we train the SVM and evaluate the performance. Sequentially, we add another feature from the list and choose the number of features where the performance curve reaches a global maximum. By proceeding in this way, the optimum number of features for our data set turned out to be 231, which spares us the calculation of more than 1,000 acoustic features.

9.7.4 Evaluation

The final acoustic anger detection unit is based on a SVM with linear kernel. In order to deal with the unbalanced class distribution, we calculate f_1 measures and use it as an evaluation criterion. The f_1 measurement is defined as the arithmetic mean of F -measures from all classes. The F -measure accounts for the harmonic mean of both precision and recall of a given class. Precision denotes what percent of class-specific predictions are correct whereas recall measures how many samples have been correctly identified as belonging to a specific class. We note that an accuracy measurement allows for false bias since it follows the majority class to a greater extent than it follows other classes. If the acoustic models follow the majority class to a greater extent, this would lead to overestimated accuracy figures. When performing tenfold cross validation⁴ with the data described in Sect. 9.5, it yields an f_1 score of 77.3%. Details are depicted in Fig. 9.6.

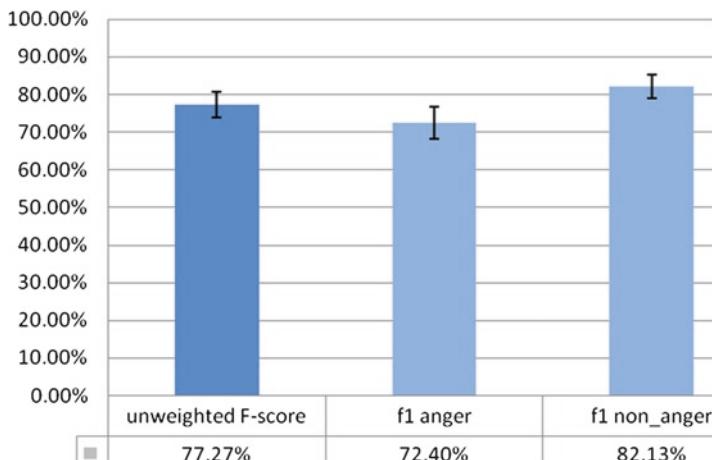


Fig. 9.6 Performance of the acoustic subsystem when evaluated with tenfold cross validation. The first bar depicts the average recognition performance of detecting the anger and the non-anger class. The second and third bar show f_1 scores for anger and non-anger

⁴The classifier is tested with one part of the set and trained with the remaining nine parts. This process is iterated ten times and the performance is averaged.

9.8 Linguistic Subsystem

For the detection of anger, loudness and intensity of the respective utterance play an important role, supporting our thesis that acoustic classifiers are expected to perform best in the task of anger detection. One would expect that linguistic information would also be of great value when detecting anger. Intuitively, we would assume that spotting swearwords would be a central task within this linguistic subsystem. However, several difficulties occurred when we wanted to exploit linguistic information: (a) the acoustically angry sounding words contained comparatively rare swearwords, (b) the speech recognizer frequently recognized swearwords that in reality never occurred, and (c) the speech recognizer was impeded in recognizing swearwords correctly because even if such expletives had been uttered by the user they would not have been recognized since those words are not included in the grammar in the first place. In sum, to be able to exploit linguistic information in anger detection, the ASR would have to be adapted to robustly recognize swearwords.

In most IVR systems, the grammar of the ASR is designed to fit the task and to deliver statistically high concept accuracy rather than return the exact word string uttered by the user. This obviously poses a problem to linguistic anger detection which would require an accurate ASR parse. A second issue in this domain is the fact that users do not necessarily employ swearwords when getting frustrated. Words like “operator” or “representative” do not appear to be particularly related to anger, but in fact they are. Yet, they certainly would never appear on any swearword list that we would design and use for keyword spotting. The challenge that is present in linguistic emotion recognition is how to find out which words users employ in a specific application domain when they become frustrated. In Lee and Narayanan [14], this problem is tackled by considering the relationship between single words and the emotion class with which they typically co-occur. Lee et al. use the term “emotional salience” in order to express the dependency of linguistic information to a certain emotion class.

The idea behind salience, in general, is that certain values in a pattern are more likely to be linked with a certain class than others. This is measured by a salience value which commensurately comes out higher the stronger the link between a concept and a class. Emotional salience considers the relationship between linguistic information and the emotion class. The assumption is that certain words are more frequently linked to distinct emotions. “Damn” would, e.g., have a rather high salience value since it co-occurs more frequently with the emotion class “anger” than with other classes, whereas “great” may have a high salience value since it more frequently co-occurs with “happiness.” Words such as “continue” or “yes” are less likely to be observed with “angry” or “happy” and thus their salience is rather low. Generally speaking, emotional salience is “a measure of the amount of information that a specific word contains about the emotion category.”

To determine the emotional salience of the contained words $w=w_1, w_2, w_3, \dots, w_n$ in the emotion classes $E=e_1, e_2, \dots, e_k$ we calculate the self-mutual information:

$$i(w_n, e_k) = \log_2 \frac{P(e_k | w_n)}{P(e_k)}.$$

$P(e_k|w_n)$ is the a posteriori probability that a word w_n co-occurs with emotion class e_k . $P(e_k)$ is the a priori probability of the emotion. If we observe a high correlation between a word w_n and an emotion class e_k then $P(e_k|w_n) > P(e_k)$. In this case, the self mutual information $i(w_n, e_k)$ is positive. If a word w_n makes an emotion class less likely, then $P(e_k|w_n) < P(e_k)$ and $i(w_n, e_k)$ is negative.

Emotional salience is defined as

$$\text{sal}(w_n) = I(E; W = w_n) = \sum_{j=1}^k P(e_j | w_n) i(w_n, e_j).$$

The mutual information I , a term from probability theory, calculates the dependency of two random variables. In this context, the mutual information I is depicted as “emotional salience.” Simply put, the higher the $\text{sal}(w_n)$ of a word, the stronger its relevance for linguistic anger detection.

Table 9.3 depicts the 20 most salient words co-occurring with the class “anger” in our speech corpus. Only words that appeared at least five times in the corpus are listed. Apart from the obvious swearwords, there are also other expressions

Table 9.3 Top 20 most salient words related to the class “anger” in the presented corpus. Listed are only words that occurred at least five times

| Word | Number in A | Number in N | $p(\text{"N"} w_n)$ | $p(\text{"A"} w_n)$ | iN | iA |
|----------------|----------------|----------------|---------------------|---------------------|-------|------|
| F*** | 0 | 5 | 0.00 | 1.00 | 0.00 | 3.14 |
| God | 1 | 7 | 0.13 | 0.88 | -2.04 | 3.00 |
| Damn | 1 | 6 | 0.14 | 0.86 | -1.90 | 2.98 |
| F***ing | 1 | 5 | 0.17 | 0.83 | -1.75 | 2.95 |
| Live | 7 | 8 | 0.47 | 0.53 | -0.72 | 2.51 |
| Person | 18 | 19 | 0.49 | 0.51 | -0.68 | 2.47 |
| Give | 5 | 5 | 0.50 | 0.50 | -0.65 | 2.44 |
| Somebody | 10 | 5 | 0.67 | 0.33 | -0.36 | 2.04 |
| Operator | 51 | 23 | 0.69 | 0.31 | -0.33 | 1.97 |
| Support | 13 | 5 | 0.72 | 0.28 | -0.28 | 1.86 |
| Talk | 26 | 10 | 0.72 | 0.28 | -0.28 | 1.86 |
| Hello | 19 | 6 | 0.76 | 0.24 | -0.23 | 1.71 |
| Representative | 79 | 24 | 0.77 | 0.23 | -0.22 | 1.68 |
| Get | 43 | 12 | 0.78 | 0.22 | -0.20 | 1.61 |
| Working | 18 | 5 | 0.78 | 0.22 | -0.20 | 1.61 |
| Please | 80 | 22 | 0.78 | 0.22 | -0.20 | 1.60 |
| Speak | 70 | 18 | 0.80 | 0.20 | -0.18 | 1.55 |
| Customer | 69 | 15 | 0.82 | 0.18 | -0.15 | 1.41 |
| That | 23 | 5 | 0.82 | 0.18 | -0.15 | 1.41 |
| Service | 81 | 14 | 0.85 | 0.15 | -0.12 | 1.22 |
| You | 52 | 8 | 0.87 | 0.13 | -0.10 | 1.12 |
| Need | 85 | 13 | 0.87 | 0.13 | -0.10 | 1.12 |
| Want | 53 | 8 | 0.87 | 0.13 | -0.10 | 1.10 |

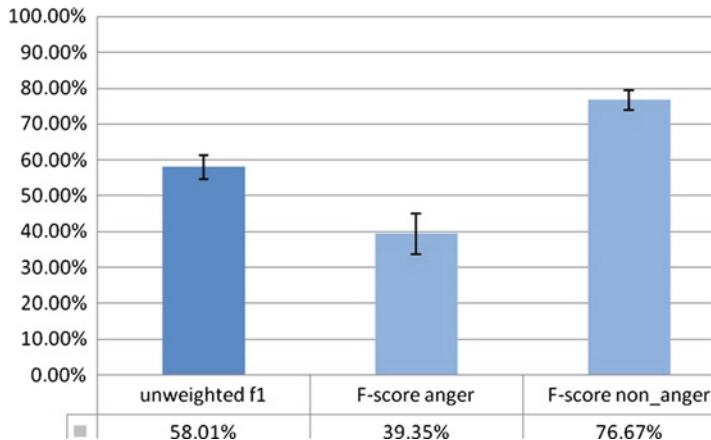


Fig. 9.7 Performance of the linguistic subsystem when evaluated with tenfold cross validation. The first bar depicts the average recognition performance of detecting the anger and the non-anger class. The second and third bar show f1 scores for anger and non-anger

carrying emotional information. In this corpus, the demand for a human operator seems to be expressed frequently when the caller is frustrated, which is indicated by words such as “operator,” “human,” “customer service,” or “representative.”

We define the probability p_A that an utterance belongs to the class “anger” as the average of all joint probabilities $P(A|w_n)$ of all words w_n :

$$p_A = \sum_{n=1}^N \frac{P(A|w_n)}{N}.$$

The subsystem that is based on a SVM, is trained with p_A as only input feature. Figure 9.7 depicts the performance of the subsystem when evaluated with tenfold cross validation.

Compared to the acoustic subsystem, the linguistic subsystem delivers a comparably poor performance. “Non-angry” user turns are, however, very well recognized notwithstanding the poor performance of the linguistic subsystem. Given that we used hand annotations of the user utterances, we assume that the ASR would deliver 100% correct parse of the user utterance. Certainly, this would not be the case in a deployed system and consequently one would have to expect a lower performance.

9.9 Call Quality Subsystem

Although we have seen that anger is an auditory and partially linguistic sensation, other factors potentially indicate that we can expect an angry user response. Frequent misrecognition from the ASR might provoke anger, inasmuch as a certain question from the system, or a sequence of questions for that matter, could be equally annoying

to the user. The fact that a user frequently interrupts the system prompts by “barging in” could imply anger, just as frequent requests for help or operator assistance.

Modern speech platforms hosting IVR systems typically allow for an extensive logging of various parameters.

Based on those parameters, we construct features that could indicate problematic dialog situations, which allow us to analyze the impact of various system factors on the caller’s emotional state. While not all features listed here necessarily have an impact on the caller’s emotional state, yet since any information is arguably good information we, therefore, collect nearly any available information and then perform IGR ranking to determine the relevant features.

9.9.1 Interaction Log Features

We initially collect parameters concerning the current dialog turn but subsequently we enhance our “window” also on the immediate context of the previous three dialog turns. We refer to those features as sliding window features. Finally, we calculate statistics on the overall course of the dialog up to this current turn that we want to classify as angry or non-angry. We call this set of features cumulative features. In the first place, we construct basic features that are logged during dialog system usage, affecting the current dialog turn between user and system. The basic features comprise information from the ASR, natural language understanding (NLU), and Dialogue Manager (DM) modules:

ASR features: the raw ASR transcription of the caller’s utterance; the ASR confidence of the returned utterance transcription; the names of all grammars active; the name of the grammar that returned the parse; did the caller begin speaking before the prompt completed (barge-in); did the caller communicate with speech (“voice”) or keypad (“DTMF”); was the speech recognizer successful (“Complete”) or not, and if it was not successful, an error message is recorded such as “NoInput”⁵ or “NoMatch.”⁶

NLU features: The semantic parse of the caller utterance as returned by the activated grammar in the current dialog module; the number of reprompts required until the NLU module succeeds in extracting a semantic meaning out of the user utterance.

Dialog Manager features: The text of what the automated agent said prior to recording the user input; the number of words of current system turn; the number of words of current user turn; the number of tries to elicit a desired response from the user; the name of the activity (aka dialog module) that is active.

⁵The NoInput event occurs when the caller does not reply to a system question within a certain time slot.

⁶The NoMatch event is triggered when the ASR is unable to recognize the user utterance with help of the activated grammars.

Sliding Window features: Operating under the assumption that it is not only the current turn that affects the user’s emotional state, but also a malfunction or behavior of the system (or user in the immediate context), we calculate sliding window features that consider the three dialog turns prior to the current one. The sliding window features count, among others, the number of NoMatch, NoInput, barge-ins in the three prior dialog steps.

Cumulative features: Operating under the assumption that a user might become frustrated due to a general malfunction of the system distributed over the complete dialog, we measure the overall call quality up to this point. For that purpose, we developed additional cumulative features analog to the sliding window features counting the number of NoMatches, NoInputs, and barge-ins up to the current dialog turn.

Ratio Features: Our ratio features take into account the percentage of misrecognitions within the ongoing dialog. We calculate them both on the cumulative values, i.e., they express what percentage of all ASR recognitions so far in the dialog delivered a NoMatch or a NoInput, and within the immediate context of the dialog, i.e., within the sliding window.

All features in detail are presented in Table 9.4.

Table 9.4 Employed contextual, non-acoustic features serving as input to the call quality sub system

| Feature | Description |
|-------------------------------|---|
| NoInput | |
| asr_cum_no_input | Cumulative number of NoInputs up to this turn |
| asr_perc_cum_no_input | Percentage of NoInputs in all previous turns |
| asr_sw_no_input | Number of NoInputs within previous three turns |
| asr_perc_sw_no_input | Percentage of NoInputs within previous three turns among all automatic speech recognition (ASR) results |
| NoMatch | |
| asr_cum_no_match | Cumulative number of NoMatchs up to this turn |
| asr_perc_cum_no_match | Percentage of NoMatchs in all previous turns |
| asr_sw_no_match | Percentage of NoMatchs within previous three turns among all ASR results |
| asr_perc_sw_no_match | Percentage of NoMatchs within previous three turns among all ASR results |
| NoInput + NoMatch | |
| asr_cum_no_input_and_no_match | Cumulative number of NoInputs + NoMatchs up to this turn |
| asr_sw_no_input_and_no_match | Number of NoInputs + NoMatchs within previous three turns |
| Success | |
| asr_cum_success | Cumulative number of successful recognitions up to this turn |
| asr_sw_success | Number of successful recognitions within previous three turns |
| BargeIn | |
| u_barged_in | True if caller barged in in current system prompt, false otherwise |
| u_cum_barged_in | Cumulative number of barge-ins up to this turn |

(continued)

Table 9.4 (continued)

| Feature | Description |
|-----------------------|---|
| u_sw_barged_in | Number of barge-ins within previous three turns |
| u_perc_cum_barged_in | Percentage of barge-ins up to this turn |
| u_perc_sw_barged_in | Percentage of barge-ins within previous three turns |
| Help_Requests | |
| nlu_cum_helpreq | Number of help requests up to this turn |
| nlu_sw_helpreq | Number of help requests within previous three turns |
| Operator_Requests | |
| nlu_cum_operatorreq | Number of operator requests up to this turn |
| nlu_sw_operatorreq | Number of operator requests within previous three turns |
| ASR_Performance | |
| asr_recognitionstatus | Was current turn successfully parsed, a NoMatch or NoInput |
| asr_confidence | ASR confidence of returned utterance transcription (0–100) |
| Other | |
| dm_activity_event | The name of the activity (aka dialog module) that is active |
| dm_prompt | Text of what automated agent said prior to recording user input 24 |
| asr_grammar_name | Names of all activated grammars |
| asr_triggered_grammar | Name of grammar that returned the ASR parse |
| u_utterance | ASR parse from user utterance |
| nlu_interpretation | Semantic parse of the caller utterance as returned grammar |
| u_utterance_duration | Duration of user utterance in seconds |
| nlu_number_of_retries | Number of re-prompts required to get desired user reply |
| dm_is_confirmation | Logs whether current system prompt is a confirmation to elicit ground b and due to ASR confidence |

9.9.2 Feature Ranking

Similar to the acoustic classifier, we will see that not all developed features will prove to be relevant for detecting anger. At this point, it would be interesting to analyze which features are indeed beneficial and which are not. The IGR ranking that we already applied in the acoustic subsystem will clarify this question.

Table 9.5 depicts the 20 most relevant features according to the IGR evaluation when performing tenfold cross validation on the corpus.

It is interesting to note that the duration of the audio file, i.e., basically the length of time of the user utterance is at the top of the table. A close relation between anger and utterance duration on the same corpus has already been shown in Schmitt et al. [22]. The class “non-angry” typically contains shorter utterances (about <2 s), whereas longer utterances tend to be “annoyed” or “hot anger” utterances. Follow-up analysis of the utterance lengths in our corpus confirms this finding: utterances labeled as angry averaged 2.07 (± 0.73) s, annoyed utterances lasted 1.82 (± 0.57) s and non-angry samples were 1.57 (± 0.66) s in average.

Generally it can be denoted that the performance of the ASR seems to have a high impact on the emotional state of the caller. The second important feature

Table 9.5 The 20 most performing features of the call quality subsystem according to information gain ratio (IGR) ranking based on entropy. The ranking has been calculated on tenfold cross validation from 2,083 user utterances (60% non-angry, 40% angry)

| | |
|----|------------------------------|
| 1 | u_utterance_duration |
| 2 | asr_sw_no_input_and_no_match |
| 3 | nlu_number_of_retries |
| 4 | asr_confidence |
| 5 | asr_sw_no_input |
| 6 | asr_perc_sw_no_input |
| 7 | nlu_interpretation |
| 8 | dm_loop_name |
| 9 | u_words_per_user_turn |
| 10 | asr_sw_success |
| 11 | nlu_cum_operatorreq |
| 12 | asr_recognition_status |
| 13 | asr_perc_sw_no_input |
| 14 | u_utterance |
| 15 | asr_perc_cum_no_match |
| 16 | u_input_mode |
| 17 | nlu_cum_operator_req |
| 18 | dm_activity_event |
| 19 | asr_perc_sw_no_input |
| 20 | asr_cum_success |

according to the ranking is the immediate number of misrecognitions in the form of NoMatch and NoInput events. This asr_sw_noinput and nomatch feature counts the number of NoMatch and NoInput events in the three previous dialog turns.

Frequent reprompts that are caused by NoMatch and NoInput events seem to annoy the caller: the number of retries (position 3) counts the number of reprompts required to elicit the desired response from the user. A high number correlates with angry callers (1.1 (non-angry) vs. 1.4 (angry)). Ranked at position 4 the asr_confidence expresses the certainty of the ASR of having recognized the correct word string. A value of below 0.5 causes the ASR to output a NoMatch event. The confidence feature is thus closely linked to the number of NoMatch events. The average confidence of the ASR when decoding a non-angry turn is 0.83 while angry turns were more difficult to recognize with a much lower confidence of 0.69. It is interesting to note that NoInput events seem to be closer related to anger than NoMatch events. We can find NoInput-related features at position 5, 6 and 13 while NoMatch-related features are less relevant at positions 15 and 19.

We also note with interest that barge-in features do not appear within the top 20. Although 46.6% of the users barge-in when being angry compared to 38.6% when being non-angry, barge-in does seem to deliver enough information to discriminate between anger and non-anger.

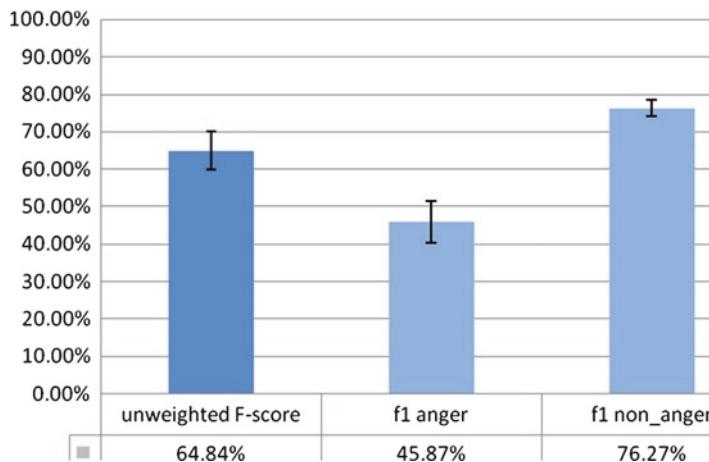


Fig. 9.8 Performance of the call quality subsystem when evaluated with tenfold cross validation. The first bar depicts the average recognition performance of detecting the anger and the non-anger class. The second and third bar show f_1 scores for anger and non-anger

9.9.3 Evaluation

The best results for the call quality subsystem could be obtained with a rule learner [6]. Details are depicted in Fig. 9.8.

While the system underperforms in predicting angry user turns, it works remarkably well in detecting non-angry turns. And given the fact that here we are working entirely without any acoustic features, the overall performance of 64.9% is more than satisfying.

9.10 Frustration History Subsystem

An important factor has been neglected so far when dealing with IVR corpora: callers do not get angry out of the clear blue sky. A certain “history of anger” can be observed as depicted in Fig. 9.9. The statistic is based on the annotated corpus that has been used for the other subsystems as well. It can be seen in the first two bars of the chart that it would be highly unlikely that a user who is non-angry in the current dialog turn had been slightly angry, or intensely angry for that matter, in the two previous turns. By contrast, it is interesting to analyze the anger history of turns where the caller showed slight or perhaps even intense anger. For example, if we observe that the user is slightly angry in the fifth dialog turn, the likelihood that the user has already been slightly angry in the two previous turns, i.e., the fourth and the third turn are 23.69 and 12.5%, respectively. In other words, when a user is angry in his current turn we have a very high probability that he will be angry in the next turn as well.

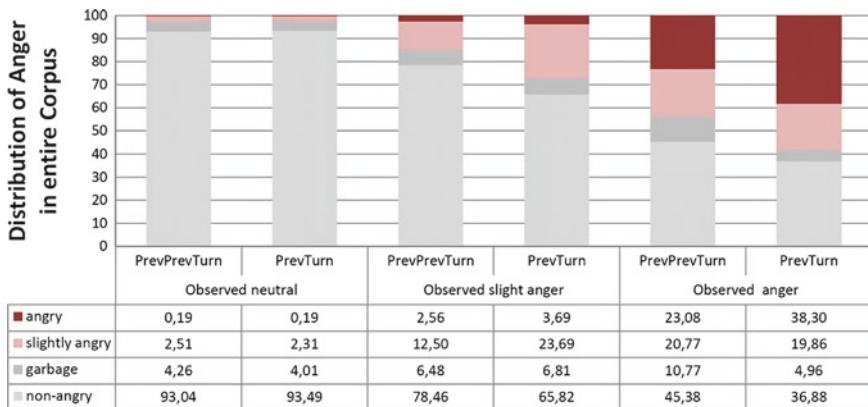


Fig. 9.9 Emotional state of the caller in percent in the two previous dialog turns of the currently considered turn. For example, when we observe anger in the current turn (see last bar), the likelihood that the caller has already been angry in the previous turn is 38%

The role of garbage turns is also striking. In this context, garbage turns are turns where the caller did engage in cross-talk or when other background noise and non-speech events such as coughing or sneezing have been recorded. When comparing non-angry and angry turns it is interesting to see that angry turns were more frequently preceded by garbage turns. The likelihood of having had a garbage turn two steps prior to a hot anger turn is even 10.77% (second last bar). Such a garbage turn inevitably leads to an ASR error which then causes the system to reprompt the question.

If we analyze the emotional state of the caller prior to the currently observed angry turn in our speech corpus, we observe sequences that might look as follows when considering the three earlier turns:

ANA
NAN
NAA
etc.

Note that “A” stands for an angry user turn and “N” stands for a non-angry user turn. Obviously, the current emotional state has a certain history of previous anger and frustration.

On the other hand, when looking at the three prior turns of a non-angry utterance, we observe much less angry turns:

NNN
NAN
NNN
etc.

In the following subsection we describe how to exploit this information and introduce our “emotion history subsystem.”

9.10.1 Hidden Markov Models for Estimating Anger

In order to model the probability of observing another angry turn (or in contrast, another non angry turn that follows non angry turns that have transpired thus far in the dialog), we train Hidden Markov models (HMM) with prior emotional states of our speech corpus. Since describing HMMs and the algorithms in detail would go beyond this chapter, we refer to Rabiner [20] where a thorough description of HMMs is provided. At this point, however, we provide a cursory description of the functionality of an HMM.

HMMs are statistical models and very popular in the domain of speech recognition, since they are able to robustly classify sequential information. Figure 9.10 depicts an HMM that models the likelihood of observing another angry turn when given a sequence of prior angry emotional states of the user. Each Markov model can be described by two random processes. The first random process consists of states (the gray circles in the figure) and the probabilities of transitioning from one state to another state. Given that the states are not visible from outside, but it is only the observation symbols or “emissions,” generated when transitioning from one state into another, that are visible, this is precisely why the term “Hidden” Markov Model is used. In this model, the observation symbols are “A” and “N.” The second random process in this framework consists of the likelihood of which observation symbol will be emitted when transitioning.

In our model, we assume that we have two states. Each time we transit from one state to the other, or decide to stay in the same state, one of the two observation symbols “A” or “N” is emitted. As can be seen in Fig. 9.10, the probability of

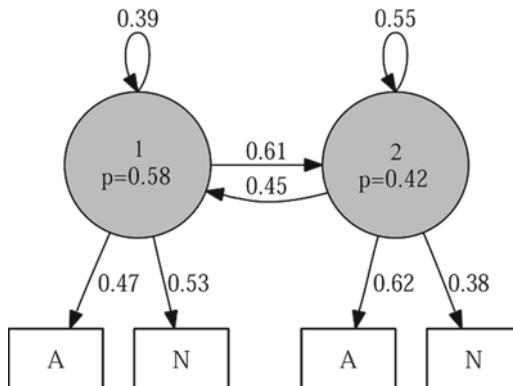


Fig. 9.10 Hidden Markov models (HMM) architecture including priors, transition and emission probabilities after training with data from the Internet agent corpus. The depicted HMM contains the probabilities for observing an angry turn

emitting an “N” is higher in the first state than in the second state. On the other hand, the probability of emitting an “A” is higher in the second state.

For each of the two possible emotional states, anger and non-anger, we trained a separate HMM with observation sequences as described above, respectively leading to anger or non-anger. During the training process, the emission and transition probabilities in each of the two HMMs are adapted.

We can now estimate the likelihood solely based on the three prior emotional states of the caller, whether the current turn is an angry or non-angry one. To achieve this, we “feed” each of the two HMMs with the observation sequence (e.g., NNA). Both HMMs calculate the probability by taking into account their respective transition and emission probabilities. The likelihood expresses how “likely it is that this specific HMM generated this observation.”

One issue still needs to be clarified: what would the subsystem predict for the first three turns in the dialog, given that the three prior emotion classes are obviously not available here. Our solution is as follows: For the third and second turn in the dialog, we proceed as described above, with the only difference that we train the HMMs only with the two or one previous turns. Thus, when classifying the first utterance, we do not use HMMs but employ the a priori probability that describes the likelihood of the currently observed utterance being an angry one based on the anger distribution in the corpus.

9.10.2 Evaluation

For evaluating the subsystem, we map the continuous probabilities of the HMMs into discrete classes. The final prediction based on the observation sequence O would be “A,” if

$$p_{\text{HMM}(A)}(O) > p_{\text{HMM}(N)}(O),$$

and “N,” if

$$p_{\text{HMM}(A)}(O) \leq p_{\text{HMM}(N)}(O).$$

Evaluation of the models is performed again with tenfold cross validation. Results are depicted in Fig. 9.11

Note that the previous emotional states “A” and “N” in this setup are not manual annotations from the user, but predictions from the acoustic subsystem. Non-angry turns yield a higher F-score than angry turns. This can be attributed to the fact that a sequence of non-angry turns (NNN) is likely to be followed by another non-angry turn. In the final classifier, we employ the continuous probabilities of the HMMs and not the discretized predictions.

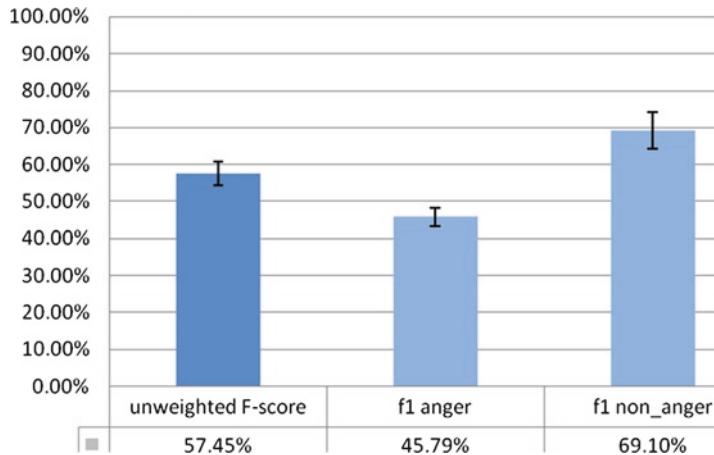


Fig. 9.11 Performance of the frustration history subsystem when evaluated with tenfold cross validation. The first bar depicts the average recognition performance of detecting the anger and the non-anger class. The second and third bar show f1 scores for anger and non-anger

9.11 Combining the Subsystems

In the previous sections, we have analyzed various information sources and their isolated contribution to anger detection. In this section, below, we analyze to what extent the subsystems actually improve the joint prediction of the complete system.

The overall prediction result of the total system is a combination of all subsystems. As a baseline, we consider the acoustic subsystem since it delivers the best performance with 77.3% F-measure among all subsystems. Multi-classifier systems (MCS) combine various classifiers. Working under the assumption that the single classifiers generate different errors, the combined result is expected to contain fewer errors.

9.11.1 Evaluation Setup

The predictions from the subsystem are used for training a simple linear perceptron serving as meta-classifier. The perceptron is trained with feature vectors containing all predictions that have the form

$$\begin{pmatrix} \text{pred}_{\text{acoustic}} \\ \text{pred}_{\text{linguistic}} \\ \text{pred}_{\text{callquality}} \\ p_{\text{history_HMM_anger}} \\ p_{\text{history_HMM_non_anger}} \end{pmatrix},$$

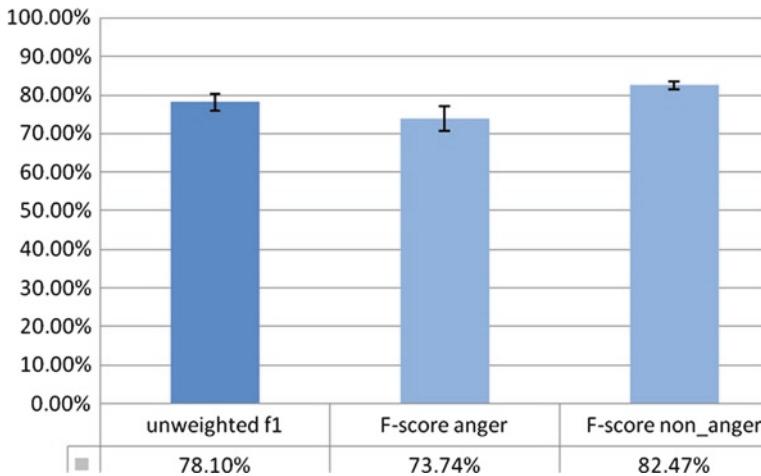


Fig. 9.12 Overall performance when evaluated with LOO validation. The first bar depicts the average recognition performance of detecting the anger and the non-anger class. The second and third bar show $f1$ scores for anger and non-anger

where the pred values contain the prediction of the respective subsystem (0 for non-angry and 1 for angry) and p the probabilities of the two HMMs of the frustration history subsystem.

The results obtained with tenfold cross validation are depicted in Fig. 9.12. The combined system yields a slightly better performance of 78.1% compared to the acoustic subsystem with 77.3%. The performance gain of 0.8%, however, is not yet satisfying. More effort has to be invested into the analysis of the errors the different subsystems generate in finding an optimum combination of all knowledge sources.

9.12 Conclusion and Discussion

Frustration in telephone-based speech applications has various aspects and we plainly see that there is more entailed in the detection of frustrated callers than acoustics alone.

In this chapter, we have analyzed four different information sources and their performance regarding the detection of angry user turns. Outperforming all other information sources in our setup, the acoustic subsystem detects both angry and non-angry user turns. Not all features extracted from the audio signal contribute to anger detection. The IGR ranking with a subsequent classification process identified roughly 230 relevant features and it turned out that loudness, intensity, and spectrals are the most relevant feature groups. That this result is also corpus dependent and can be generalized only to a certain extent has been shown in [18], where the corpus employed in this work is compared with a German IVR corpus. One might expect that linguistic information substantially adds to the level of performance of an anger

detection task. This might be the case when trying to detect callers that are in a state of rage. Remember, however, that we combined angry and annoyed user turns in this classification task, and the overall amount of hot angry user turns in the complete corpus amounted to less than 1%. Linguistically, since we have seen that annoyed utterances uncannily resemble non-angry utterances, drawing a distinction merely based on linguistics would be difficult. Adding to this difficulty is the fact that IVR data contains mostly short utterances or perhaps single words only like “yes,” “no,” or “continue” which are naturally too general to indicate one of the two emotion classes even though these words are used in both cases. Other studies report a higher performance when using emotional salience [19].

Aspects of call quality have been modeled in the call quality subsystem that makes use of interaction parameters, which indicate, among other things, the performance of the speech recognizer, the barge-in behavior of the caller, the number of help and operator requests from the user, and so on. An analysis of the relevant features of the subsystem disclosed that particularly NoInput events have a close link to the detection of anger. Surprisingly, the barge-in behavior of the caller is less relevant for anger detection.

The development of anger over time has been illuminated and exploited within the frustration history subsystem. Under the assumption that angry turns frequently co-occur with other angry turns, a model has been presented based on HMMs that estimates the likelihood of observing other angry turns after previously spotting angry turns. Again, the distinction of non-angry turns was more robust than the distinction of angry turns.

The combined performance of the system yields 78.1% when employing a voted perceptron as meta-classifier which is a slight improvement of 0.8% compared to the best subsystem, the acoustic classifier. In future work, we will analyze the optimum combination of the subsystems and consider the respective strengths and weaknesses of each of the single units in more detail.

References

1. Acomb, K., Bloom, J., Dayanidhi, K., Hunter, P., Krogh, P., Levin, E., and Pieraccini, R. (2007). Technical support dialog systems: issues, problems, and solutions. In Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies, pages 25–31. Rochester, NY: Association for Computational Linguistics.
2. Batliner, A., Fischer, K., Huber, R., Spilker, J., and Nöth, E. (2000). Desperately seeking emotions: actors, wizards, and human beings. In Cowie, R., Douglas-Cowie, E., and Schröder, M., editors, Proceedings of the ISCA Workshop on Speech and Emotion, pages 195–200.
3. Bennett, K. P. and Campbell, C. (2000). Support vector machines: hype or hallelujah? Journal of SIGKDD Explorations, 2(2):1–13.
4. Bizacumen Inc. (2009). Interactive voice response (IVR) systems – an international market report. Market study, Bizacumen Inc.
5. Burkhardt, F., Rolfes, M., Sendlmeier, W., and Weiss, B. (2005). A database of German emotional speech. In Proceedings of the International Conference on Speech and Language Processing (ICSLP) Interspeech 2005, ISCA, pages 1517–1520.

6. Cohen, W. W. and Singer, Y. (1999). A simple, fast, and effective rule learner. In Proceedings of the 16th National Conference on Artificial Intelligence, pages 335–342. Menlo Park, CA: AAAI Press.
7. Davies, M. and Fleiss, J. (1982). Measuring agreement for multinomial data. *Biometrics*, 38:1047–1051.
8. Enberg, I. S. and Hansen, A. V. (1996). Documentation of the Danish emotional speech database. Technical report, Aalborg University, Denmark.
9. Gorin, A. L., Riccardi, G., and Wright, J. H. (1997). How may I help you? *Journal of Speech Communication*, 23(1–2):113–127.
10. Herm, O., Schmitt, A., and Liscombe, J. (2008). When calls go wrong: how to detect problematic calls based on log-files and emotions? In Proceedings of the International Conference on Speech and Language Processing (ICSLP) Interspeech 2008, pages 463–466.
11. Kim, W. (2007). Online call quality monitoring for automating agentbased call centers. In Proceedings of the International Conference on Speech and Language Processing (ICSLP).
12. Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.
13. Langkilde, I., Walker, M., Wright, J., Gorin, A., and Litman, D. (1999). Automatic prediction of problematic human-computer dialogues in how may I help you. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU99, pages 369–372.
14. Lee, C. M. and Narayanan, S. S. (2005). Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2):293–303.
15. Levin, E. and Pieraccini, R. (2006). Value-based optimal decision for dialog systems. In Proceedings of Spoken Language Technology Workshop 2006, pages 198–201.
16. Paek, T. and Horvitz, E. (2004). Optimizing automated call routing by integrating spoken dialog models with queuing models. In HLT-NAACL, pages 41–48.
17. Pieraccini, R. and Huerta, J. (2005). Where do we go from here? Research and commercial spoken dialog systems. In Proceedings of the 6th SIGdial Workshop on Discourse and Dialog, pages 1–10.
18. Polzehl, T., Schmitt, A., and Metze, F. (2009). Comparing features for acoustic anger classification in German and English IVR portals. In First International Workshop on Spoken Dialogue Systems (IWSDS).
19. Polzehl, T., Sundaram, S., Katabdar, H., Wagner, M., and Metze, F. (2009). Emotion classification in children’s speech using fusion of acoustic and linguistic features. In Proceedings of the International Conference on Speech and Language Processing (ICSLP) Interspeech 2009.
20. Rabiner, L. R. (1990). A tutorial on hidden Markov models and selected applications in speech recognition. San Francisco, CA: Morgan Kaufmann.
21. Schmitt, A., Hank, C., and Liscombe, J. (2008). Detecting problematic calls with automated agents. In 4th IEEE Tutorial and Research Workshop Perception and Interactive Technologies for Speech-Based Systems, Irsee, Germany.
22. Schmitt, A., Heinroth, T., and Liscombe, J. (2009). On nomatchs, noinputs and barge-ins: do non-acoustic features support anger detection? In Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue, SigDial Conference 2009, London, UK: Association for Computational Linguistics.
23. Schuller, B. (2006). Automatische Emotionserkennung aus sprachlicher und manueller Interaktion. Dissertation, Technische Universität München, München.
24. van den Bosch, A., Krahmer, E., and Swerts, M. (2001). Detecting problematic turns in human-machine interactions: rule-induction versus memory-based learning approaches. In ACL’01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pages 82–89, Morristown, NJ: Association for Computational Linguistics.
25. Walker, M. A., Langkilde-Geary, I., Hastie, H. W., Wright, J., and Gorin, A. (2002). Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, 16:293–319.
26. Zweig, G., Siohan, O., Saon, G., Ramabhadran, B., Povey, D., Mangu, L., and Kingsbury, B. (2006). Automated quality monitoring in the call center with ASR and maximum entropy. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2006, volume 1, pages 1–12.

Chapter 10

“The Truth is Out There”: Using Advanced Speech Analytics to Learn Why Customers Call Help-line Desks and How Effectively They Are Being Served by the Call Center Agent

Marsal Gavalda and Jeff Schlueter

Abstract In this chapter, we describe our novel work in phonetic-based indexing and search, which is designed for extremely fast searching through vast amounts of media. This method makes it possible to search for words, phrases, jargon, slang, and other terminology that are not readily found in a speech-to-text dictionary. The most advanced phonetic-based speech analytics solutions, such as ours, are those that are robust to noisy channel conditions and dialectal variations; those that can extract information beyond words and phrases; and those that do not require the creation or maintenance of lexicons or language models. Such well-performing speech analytic programs offer unprecedented levels of accuracy, scale, ease of deployment, and an overall effectiveness in the mining of live and recorded calls. Given that speech analytics has become sine qua non to understanding how to achieve a high rate of customer satisfaction and cost containment, we demonstrate in this chapter how our data mining technology is used to produce sophisticated analyses and reports (including visualizations of call category trends and correlations or statistical metrics), while preserving the ability at any time to drill down to individual calls and listen to the specific evidence that supports the particular categorization or data point in question, all of which allows for a deep and fact-based understanding of contact center dynamics.

Keywords Audio data mining • Audio search • Speech analytics • Customer satisfaction • Live and recorded calls • Call category trends • Contact centers • Digital audio and video files • Phonetic indexing and search • Average handle time

M. Gavalda (✉)

Vice President of Incubation and Principal Language Scientist,
Nexidia, 3565 Piedmont Road, NE, Building Two,
Suite 400, Atlanta, GA 30305, USA
e-mail: renee@philosophypr.com

10.1 Introduction

Speech analytics is a new field that applies speech and language technologies to transform unstructured audio into business intelligence and provides solutions for commercial contact centers, government intelligence, legal discovery and rich media, among many applications.

From contact centers to broadcast news to podcasts, the quantity of digital audio and video files being created is growing quickly and shows no signs of slowing. While valuable information for the enterprise is contained in these files, there has historically been no effective means to organize, search, and analyze the data in an efficient manner.

Consequently, much of these data were unavailable for analysis, such as the millions of hours of contact center calls recorded every year for quality control. Using a traditional approach, a very small amount of audio may be listened to, but in an ad-hoc manner, such as random audits by contact center managers, or manual monitoring of various broadcasts. Targeted searching, however, is difficult. If this audio data were easily searchable, many applications would be possible, such as reviewing only calls that meet certain criteria, performing trend analysis across thousands of hours of customer calls, searching a collection of newscasts to find the exact locations where a certain topic is discussed, among many other uses.

The difficulty in accessing the information in most audio today is that unlike most broadcast media, closed captioning is not available. Further, man-made transcripts are expensive to generate, and limited in their description. Audio search based on speech-to-text technology is not scalable and depends on customized dictionaries and acoustic models, which translates into a prohibitive total cost of ownership. What is needed is another approach.

In this chapter, we summarize prior work in searching audio data, and examine the salient characteristics of various traditional methods. We then introduce and describe a different approach known as phonetic-based indexing and search, designed for extremely fast searching through vast amounts of media which allows the search for words, phrases, jargon, slang, and other terminology not readily found in a speech-to-text dictionary. After explaining the current applications of speech analytics in the contact center market, we end with a look at next generation technologies such as real-time monitoring and detection of emerging trends with cross-channel analytics.

10.2 History of Audio Search

10.2.1 Prior Work, Approaches and Techniques

Retrieval of information from audio and speech has been a goal of many researchers over the past 50 years (see Fig. 10.1), where the main historical trends have been to increase the vocabulary and move away from isolated word and speaker-dependent

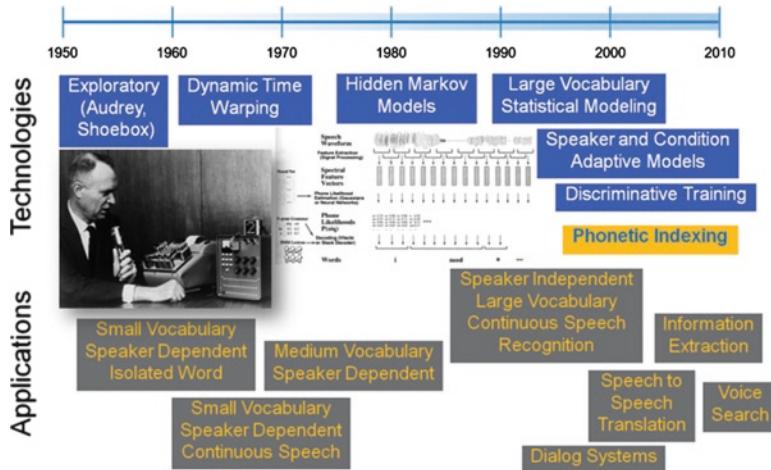


Fig. 10.1 A view of the history of speech recognition technologies and applications (adapted from Ref. [2])

recognition. The simplest solution to spoken content analysis would be to use large vocabulary continuous speech recognition (LVCSR), perform time alignment, and produce an index of text content along with time stamps. LVCSR is sufficiently mature that toolboxes are publicly available such as HTK (from Cambridge University, England), ISIP (Mississippi State University, USA), and Sphinx (Carnegie Mellon University, USA) as well as a host of commercial offerings. Much of the improved performance demonstrated in current LVCSR systems comes from better linguistic modeling [11] to eliminate sequences of words that are not allowed within the language. Unfortunately, the word error rates are very high (in the 40–50% range for typical contact center data).

The need for better automatic retrieval of audio data has prompted formulation of databases specifically to test this capability [8]. Also, a separate track has been established for spoken document retrieval within the annual TREC (Text Retrieval Conference) event [6]. An example can be seen in Ref. [10]. In this research, a transcription from LVCSR was produced on the NIST-sponsored HUB-4 Broadcast News corpus. Whole sentence queries are posed, and the transcription is searched using intelligent text-based information extraction methods. Some interesting results from this report show that word error rates range from 64% to 20%, depending on the LVCSR system used, and closed captioning error rates are roughly 12%. While speech recognition has improved since these results, the improvement has been measured and modest. For example, 10 years after the TREC project was initiated (and stayed) in the comparatively easy domain of broadcast news recordings, recent work by Google’s Speech Research Group [1] indicates that, even after tuning an LVCSR system by adding 7.7 million words to train the language model, 6,000 words to the lexicon and “manually checking and correcting” the pronunciations of the most frequent ones, the performance of the resulting system was less

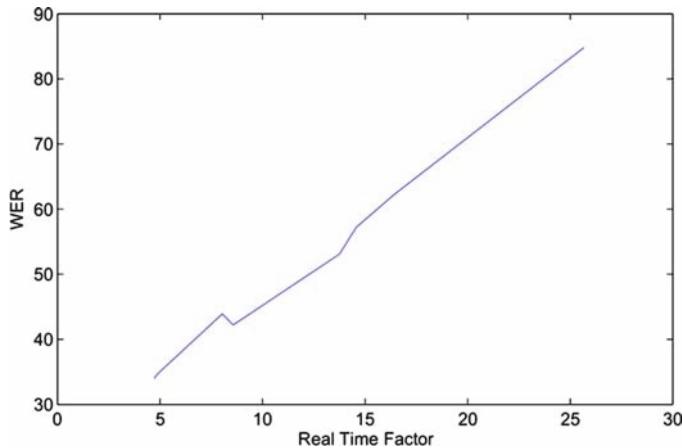


Fig. 10.2 LVCSR systems experience a severe tradeoff between speed and accuracy. Here, a broadcast news engine is sped up from 5 to $25 \times RT$, propelling the word error rate from the mid-1930s to the high 1980s [13]

than stellar: a word error rate of 36.4% and an out-of-vocabulary rate of 0.5% all the while the engine is running at less than real time ($0.77 \times RT$). Additionally, if an LVCSR system is sped up by decreasing the beam width, for example, a severe penalty is incurred in the form of very high word error rates (see Fig. 10.2).

In the LVCSR approach, the recognizer tries to transcribe all input speech as a chain of words in its vocabulary. Keyword spotting is a different technique for searching audio for specific words and phrases. In this approach, the recognizer is only concerned with occurrences of one keyword or phrase. Since the score of the single word must be computed (instead of the entire vocabulary), much less computation is required, which was important for early real-time applications such as surveillance and automation of operator-assisted calls [15, 16]. Also Ng and Zue [12] recognized the need for phonetic searching by using subword units for information retrieval, except that the reported phonetic error rates were high (37%) and performance of the retrieval task was low compared to LVCSR methods.

10.2.2 *Invention of Phonetic Indexing and Search*

In the mid-1990s, a different approach to spoken content analysis was developed: phonetic indexing and search. It differs from LVCSR in the sense that it does not attempt to create a transcript, but rather generates a phonetic index (also known as phonetic search track) against which searches are performed. As illustrated in Fig. 10.3, and described in more detail in Refs. [3–5], phonetic-based search comprises two phases: indexing and searching.

The first phase, which indexes the input speech to produce a phonetic search track, is performed only once. The second phase is to search the phonetic track,

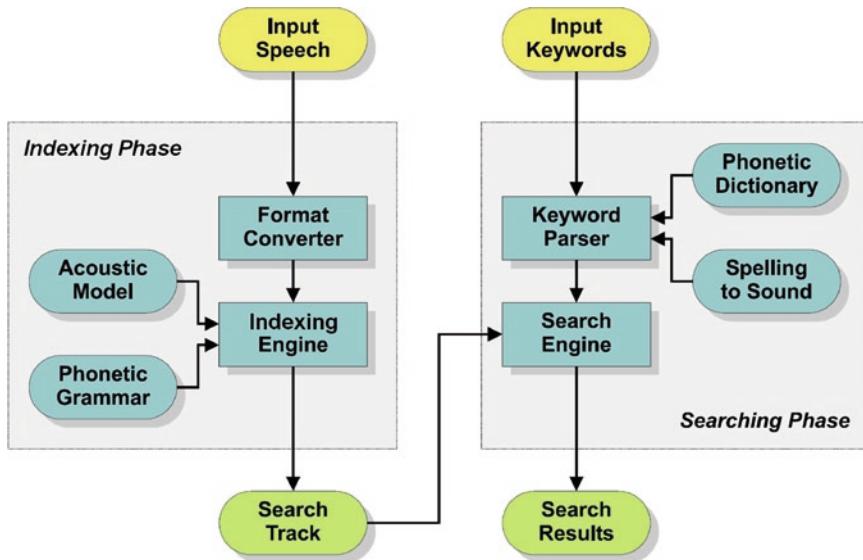


Fig. 10.3 Logical architecture of phonetic indexing and search

performed whenever a search is needed for a word or phrase. Once the indexing is completed, this search stage can be repeated for any number of queries. Since the search is phonetic, search queries do not need to be in any predefined dictionary, thus allowing searches for proper names, new words, misspelled words, jargon, etc. Note that once indexing has been completed, the original media are not involved at all during searching. Thus, the search track can be generated from the highest-quality media available for improved accuracy (e.g., µ-law audio for telephony), but this audio can then be replaced by a compressed representation for storage and subsequent playback (e.g., GSM) afterwards.

The indexing phase begins with format conversion of the input media (whose format might be MP3, ADPCM, QuickTime, etc.) into a standard audio representation for subsequent handling (PCM). Then, using an acoustic model, the indexing engine scans the input speech and produces the corresponding phonetic search track. An acoustic model jointly represents characteristics of both an acoustic channel (an environment in which the speech was uttered and the transducer through which it was recorded) and a natural language (in which human beings expressed the input speech). Audio channel characteristics include: frequency response, background noise, and reverberation. Characteristics of a natural language include gender, dialect, and accent of the speaker. Typically at least two acoustic models are produced for each language: a model for media with higher sampling rates, good signal-to-noise ratios, and more formal, rehearsed speech; and a model for media from a commercial telephony network, either landline or cellular handset, optimized for the more spontaneous, conversational speech of telephone calls.

The end result of phonetic indexing of an audio file is a highly compressed representation of the phonetic content of the input speech (see Figs. 10.4–10.6).



Fig. 10.4 Example of a snippet of digitized speech, drawn as a waveform. A waveform is a plot of the energy of the signal across time, where the horizontal axis represents time (2 s in the fragment plotted) and the vertical axis represents the loudness of the sound (plotted here from $-\infty$ to -15 dB)

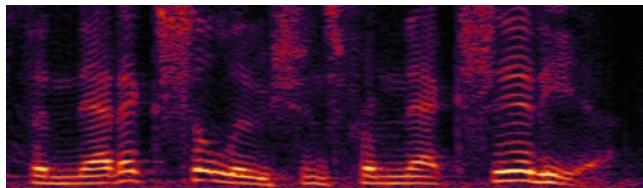


Fig. 10.5 Spectral analysis of the speech snippet in Fig. 10.4 in the form of a spectrogram. A spectrogram is a plot of the distribution of frequencies across time, i.e., how the energy of each frequency band changes over time. The horizontal axis represents time (2 s in this case), the vertical axis represents frequencies (shown here from 0 to 4,000 Hz), and the intensity of color (yellow for peaks, black for valleys) corresponds to the energy of each frequency band at each point in time



Fig. 10.6 Representation of a search result against a phonetic index obtained for the speech snippet in Figs. 10.4 and 10.5. Phonemes are represented by boxes, where the length of the box corresponds to the duration of the phoneme and the color of the box represents how well the speech signal matches the model along a gradient from red (poor match) to green (good match). The sequence of phonemes for this example is /f ah n eh t ih k s ah l uw sh ah n z f r ah m n eh k s ih d iy ah/ (i.e., “phonetic solutions from Nexidia”)

Unlike LVCSR, whose essential purpose is to make irreversible (and possibly incorrect) bindings between speech sounds and specific words, phonetic indexing merely infers the likelihood of potential phonetic content as a reduced lattice, deferring decisions about word bindings to the subsequent searching phase.

The searching phase begins with parsing the query string, which is specified as text containing one or more:

- Words or phrases (e.g., “president” or “supreme court justice”)
- Phonetic strings (e.g., “_eh _m _p _iy _th _r _iy,” seven phonemes representing the acronym “MP3”)
- Temporal operators (e.g., “brain development &15 bisphenol A,” representing two phrases spoken within 15 s of each other)

A phonetic dictionary is referenced for each word within the query term to accommodate *unusual* words (such as those whose pronunciations must be handled specially for the given natural language) as well as very *common* words, for which performance optimization is worthwhile. Any word not found in the dictionary is

then processed by consulting a spelling-to-sound converter that generates likely phonetic representations given the word’s orthography.

Multiple index files can be scanned at high speed during a single search for likely phonetic sequences (possibly separated by offsets specified by temporal operators) that closely match corresponding strings of phonemes in the query term. Recall that index files encode potential sets of phonemes, not irreversible bindings to sounds. Thus, the matching algorithm is probabilistic and returns multiple results, each as a 4-tuple:

- Index File (to identify the media segment associated with the putative hit)
- Start Time Offset (beginning of the query term within the media segment, accurate to one hundredth of a second)
- End Time Offset (approximate time offset for the end of the query term)
- Confidence Level (that the query term occurs as indicated, between 0.0 and 1.0)

Even during searching, irreversible decisions are postponed. Results are simply enumerated, sorted by confidence level, with the most likely candidates listed first. Post processing of the results list can be automated. Example strategies include hard thresholds (e.g., ignore results below 90% confidence), occurrence counting (e.g., a media segment gets a better score for every additional instance of the query term), and natural language processing (patterns of nearby words and phrases denoting semantics).

Typical web search engines strive to return multiple results on the first page so that the user can quickly identify one of the results as their desired choice. Similarly, an efficient user interface can be devised to sequence rapidly through a phonetic search results list, to listen briefly to each entry, to determine relevance and, finally, to select one or more utterances that meet specific criteria. Depending on available time and importance of the retrieval, the list can be perused as deeply as necessary.

10.2.3 Structured Queries

In addition to ad-hoc searches, more complex queries commonly known as “structured queries” are needed to better model the context of what needs to be captured. A structured query is similar to a finite-state grammar produced for an automatic speech recognition system. Examples of operators are AND, OR, BEFORE, SUBSET, ANDNOT, FIRST, LAST, etc. Due to the special domain of audio search, several helpful extensions are also provided, such as attaching time windows to operators. By constructing complex queries, analysts are able to classify calls by call driver, customer sentiment, and so forth, in addition to detecting word or word phrase occurrences only. An example might be that of identifying how many calls in a contact center’s archive discuss problems with a rebate? Structured queries are simple to write and yet they have the expressive power to capture complex Boolean and temporal relationships, as shown in the following example:

- Satisfaction = **OR**(“how did you like,” “are you happy with,” “how was your experience”)
- Negative = **OR**(“not satisfied,” “terrible experience,” “negative feedback”)

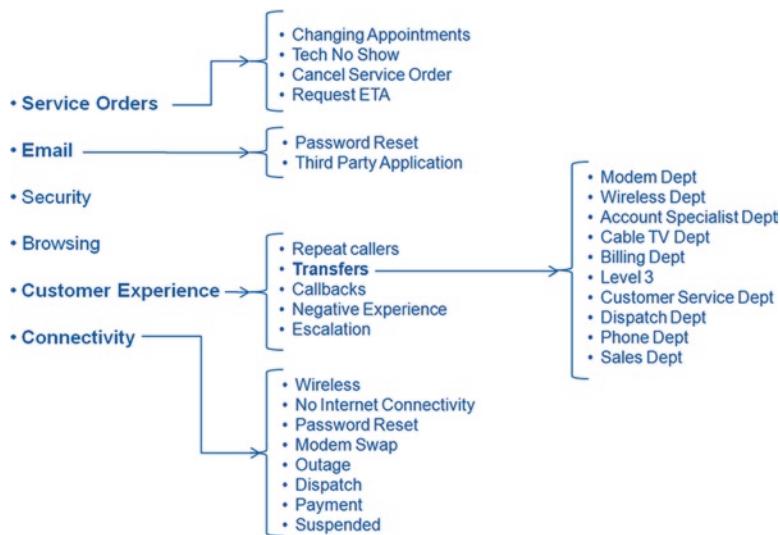


Fig. 10.7 Example of a three-level call driver taxonomy developed for a telecommunications company. Each category is captured by a structured query (combination of phrases via Boolean and time-based operators)

- NegativeSatisfaction = **BEFORE_10**(Satisfaction, Negative)
- Query = **LAST_180**(NegativeSatisfaction)

It is through structured queries that the kind of call categorization shown in Fig. 10.7 is achieved.

10.3 Advantages of Phonetic Search

The basic architecture of phonetic searching offers several key advantages over LVCSR and conventional word spotting:

- *Speed, accuracy, scalability.* The indexing phase devotes its limited time allotment purely to categorizing input speech sounds into potential sets of phonemes - rather than making irreversible decisions about words. This approach preserves the possibility for high accuracy speech recognition, enabling the searching phase to make better decisions when presented with specific query terms. Also, the architecture separates indexing and searching so that the indexing needs to be performed only once, while the relatively fast operation (searching) can be performed as often as necessary.
- *Open vocabulary.* LVCSR systems can only recognize words found in their lexicons. Many common query terms (such as specialized terminology and names of people, places and organizations) are typically omitted from these lexicons (partly to keep them small enough that LVCSRs can be executed cost-effectively in real time, and also because these kinds of query terms are

- notably unstable as new terminology and names are constantly evolving). Phonetic indexing is unconcerned about such linguistic issues, maintaining completely open vocabulary (or, perhaps more accurately, no vocabulary at all).
- *Low penalty for new words.* LVCSR lexicons can be updated with new terminology, names, and other words. However, this exacts a serious penalty in terms of cost of ownership - because the entire media archive must then be reprocessed through LVCSR to recognize the new words (an operation that typically executes only slightly faster than real time at best). Also, probabilities need to be assigned to the new words, either by guessing their frequency or context or by retraining a language model that includes the new words. The dictionary within the phonetic searching architecture, on the other hand, is consulted only during the searching phase, which is relatively fast when compared to indexing. Adding new words incurs only another search, and it is seldom necessary to add words, since the spelling-to-sound engine can handle most cases automatically, or, if not, users can simply enter sound-it-out versions of words.
 - *Phonetic and inexact spelling.* Proper names are particularly useful query terms - but also particularly difficult for LVCSR, not only because they may not occur in the lexicon as described above, but also because they often have multiple spellings (and any variant may be specified at search time). With phonetic searching, exact spelling is not required. For example, “Sudetenland” could also be searched for as “Sue Dayton Land.” This advantage becomes clear with a name that can be spelled “Qaddafi,” “Khaddafi,” “Quadafy,” “Kaddafi,” or “Kadoffee” - any of which could be successfully located by phonetic searching.
 - *User-determined confidence threshold.* If a particular word or phrase is not spoken clearly, or if background noise interferes at that moment, then LVCSR will likely not recognize the sounds correctly. Once that decision is made, the correct interpretation is hopelessly lost to subsequent searches. Phonetic searching however returns multiple results, which are sorted by confidence level. The sounds at issue may not be the first (it may not even be in the top ten or 100), but it is very likely in the results list somewhere, particularly if some portion of the word or phrase is relatively unimpeded by channel artifacts. If enough time is available, and if the retrieval is sufficiently important, then a motivated user (aided by an efficient human interface) can drill as deeply as necessary.
 - *Amenable to parallel execution.* The phonetic searching architecture can take full advantage of any parallel processing accommodations. For example, a computer with four processors can index four times as fast. Additionally, PAT files can be processed in parallel by banks of computers to search more media per unit time.

10.3.1 Performance of Phonetic Search

Phonetic-based solutions are designed to provide high performance for both indexing and search. The engine is designed to take maximum advantage of a multi-processor system, such that a dual processor box achieves nearly double the throughput of a single processor configuration, with minimal overhead between

processors. Compared to alternative LVCSR approaches, the phonetic-based search engine provides a level of scalability not achievable by other systems.

Typically, the engine comes with built-in support for a wide variety of common audio formats, including PCM, μ -law, A-law, ADPCM, MP3, QuickTime, WMA, G.723.1, G.729, G.726, Dialogic VOX, GSM and many others, as well as a framework to support custom file-formats and devices, such as direct network feeds and proprietary codecs, through a plug-in architecture.

There are three key performance characteristics of phonetic search: accuracy of results, index speed, and search speed. All three are important when evaluating any audio search technology. This section will describe each of these in detail.

10.3.2 Measuring Accuracy

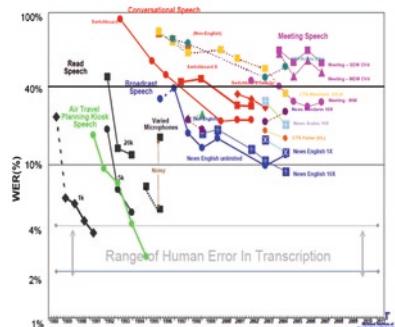
Measuring the accuracy of a phonetic-based indexing and search system requires different metrics from speech-to-text (see Fig. 10.8). Rather than a single value such as word error rate, accuracy of a phonetic-based search is measured by precision and recall, as for any other information retrieval task. Phonetic-based search results are returned as a list of putative hit locations, in descending likelihood order. That is, as users progress further down this list, they will find the occurrence of more and more

Speech to Text

Accuracy based on correctness of output transcript

Word error rate: number of substitutions + deletions + insertions, divided by number of actual words

| | | |
|------------------|--------------------------------------|-------|
| REF | you weren't born just to soak up sun | |
| HYP ₁ | you weren't born justice see cups on | [57%] |
| HYP ₂ | you weren't born just to sew cups on | |
| HYP ₃ | you weren't born justice vocal song | [43%] |



Phonetic Indexing and Search

Accuracy based on precision and recall

Precision: number of correct hits above threshold, divided by total number of hits above threshold

Recall: number of correct hits above threshold, divided by total number of correct hits (independent of threshold)

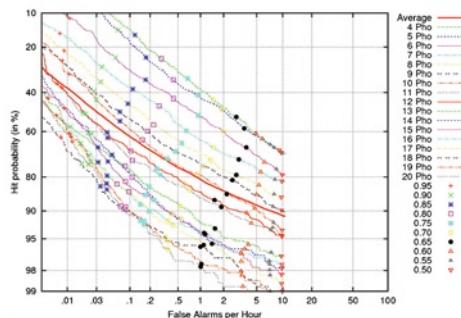


Fig. 10.8 Measuring accuracy in speech-to-text and phonetic indexing (adapted from Ref. [7]). On the left is a chart depicting (in a non-linear scale) the word error rates for a variety of speech-to-text tasks, ranging in difficulty from read speech (black) to spontaneous, conversational, multi-speaker meeting conversations (pink). On the right is a decision-error-tradeoff plot showing how, for a phrase detection task based on phonetic indexing, recall increases (top-to-bottom vertical axis) as false alarms per hour increase (left-to-right horizontal axis). For example, with a tolerance of two false alarms per hour, a 19-phoneme phrase obtains a 98% recall

instances of their query. However, they will also eventually encounter an increasing amount of false alarms (results that do not correspond to the desired search term). This performance characteristic is best shown by a curve common in detection theory: the receiver operating characteristic (ROC) curve, shown in Fig. 10.9 (or the equivalent decision error tradeoff curve as shown in Fig. 10.8).

To generate these curves, one needs experimental results from the search engine (the ordered list of putative hits) and the ideal results for the test set (acquired by manual review and documentation of the test data). For audio search, the ideal set is the verbatim transcripts of what was spoken in the audio. For a single query, the number of actual occurrences in the ideal transcript is counted first. The ROC curve begins at the 0,0 point on graph of False Alarms per Hour versus Probability of Detection. Results from the search engine are then examined, beginning from the top of the list. When a putative hit in the list matches the transcript, the detection rate increases, as the percentage of the true occurrences detected has just gone up (the curve goes up). When the hit is not a match, the false alarm rate now increases (the curve now moves to the right). This continues until the false alarm rate reaches a predefined threshold. For any single query in generic speech, this curve normally has very few points, since the same phrase will only happen a few times, unless the same topic is being discussed over and over in the database. To produce a meaningful ROC curve, thousands of queries are tested with the results averaged together, generating smooth, and statistically significant, ROC curves.

There are two major characteristics that affect the probability of detection of any given query: the type of audio being searched; and the length and phoneme composition of the search terms themselves.

To address the first issue, two language packs for each language are typically provided, one designed to search broadcast-quality media and another for telephony-quality audio. The ROC curves for North American English in broadcast and telephony are shown in Fig. 10.9.

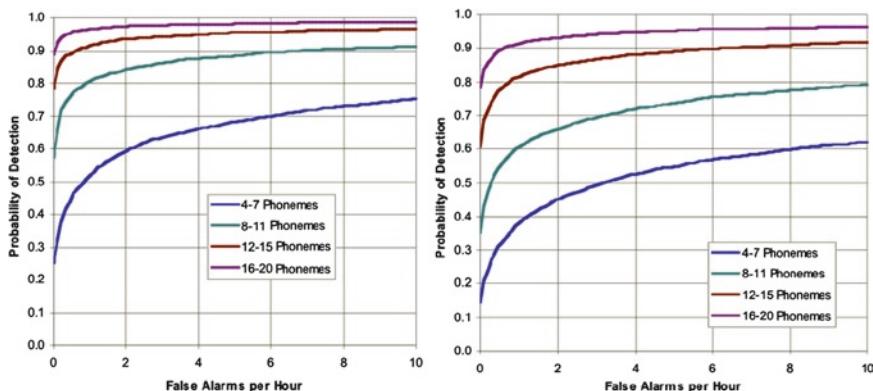


Fig. 10.9 Sample ROC curves for North American English for the broadcast (left chart) and telephony (right chart) language packs

For example, using the North American English broadcast language pack and a query length of 12–15 phonemes, you can expect, on average, to find 85% of the true occurrences, with less than one false hit per 2 h of media searched.

In a word-spotting system, more phonemes in the query mean more discriminative information is available at search time. As shown by the four curves in the charts representing four different groups of query lengths, the difference can be dramatic. Fortunately, rather than short, single word queries (such as “no” or “the”), most real-world searches are for proper names, phrases, or other interesting speech that represent longer phoneme sequences.

10.3.3 Indexing Speed

Another significant metric of phonetic systems is indexing speed (i.e., speed at which new media can be made searchable). This is a clear advantage for phonetic-based solutions, as the engine ingests media very rapidly. From contact centers with hundreds of seats, media archives with tens of thousands of hours, or handheld devices with limited CPU and power resources, this speed is a primary concern, as this relates directly to infrastructure cost (see Fig. 10.10).

Indexing requires a relatively constant amount of computation per media hour, unless a particular audio segment is mostly silence, in which case the indexing rates are even greater. In the worst-case scenario of a contact center or broadcast recording that contains mostly non-silence, index speeds for a server-class PC are given below in Table 10.1.

These speeds indicate that the indexing time for 1,000 h of media is less than 1 h of real time. Put another way, a single server at full capacity can index over



Fig. 10.10 Scalability comparison of speech-to-text (at $4 \times$ RT) and phonetic indexing (at $207 \times$ RT)

Table 10.1 Typical index speeds in times faster than real time for a 12-phoneme search term on a 2-processor, 4-core server

| Index speed ($\times RT$) | Server utilization |
|-----------------------------|---|
| 207 | 12.5% (single thread, only 1 CPU core used) |
| 1,457 | 100% (8 threads, one thread per CPU core) |

30,000 h of media per day. These results are for audio supplied in linear PCM or μ -law format to the indexing engine. If the audio is supplied in another format such as MP3, WMA, or GSM, there will be a small amount of format-dependent overhead to decode the compressed audio.

10.3.4 Search Speed

A final performance measure is the speed at which media can be searched once it has been indexed. Two main factors influence the speed of searching. The most important factor is whether the phonetic indices are in memory or on disk. Once an application requests a search track to be loaded, the search engine will load the track into memory. Any subsequent searching will use this in-memory version, a process which serves to speed up the process significantly when the same media is searched multiple times.

A second factor influencing search speed is the length, in phonemes, of the word or phrase in the query (see Fig. 10.11). Shorter queries run faster, as there are fewer calculations to make internal to the search engine.

Table 10.2 below shows the search speeds for a fairly average (12 phonemes long) query over a large set of in-memory index files, executed on a server-class PC.

10.3.5 Additional Information Extracted from Audio

In addition to the creation of a phonetic index to support the analysis of spoken content, other types of information can be extracted from the audio that are highly relevant for contact center analytics. They include:

- *Voice activity detection:* In order to determine the amount of “dead air” time on a call, e.g., when a customer is placed on hold, it is important to detect when speech occurs. Typically, pauses longer than 7 s are considered non-talk time for reporting purposes.
- *Language ID:* Language family and specific language ID can be computed for a call, even locating specific segments within a call where a particular language is being spoken.
- *DTMF:* Touch tone numbers can be extracted from a recording and added as metadata fields.
- *Music and gender:* Other detectors such as music and gender can be run against the audio.

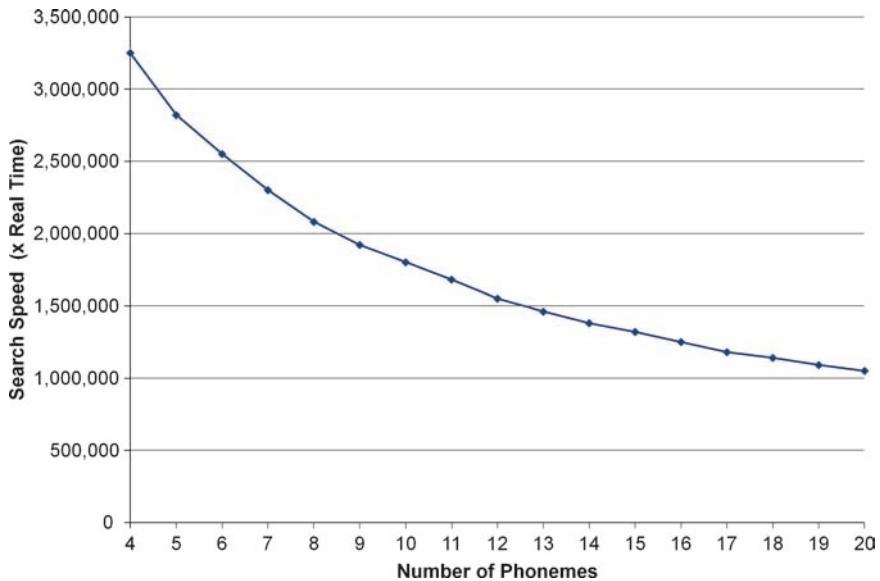


Fig. 10.11 Search speed, in hours of media searched per second of CPU time, for a range of search term lengths

Table 10.2 Typical search speeds in times faster than real time for a 12-phoneme search term on a 2-processor, 4-core server

| Search speed (xRT) | Server utilization |
|--------------------|---|
| 667,210 | 12.5% (single thread, only 1 CPU core used) |
| 5,068,783 | 100% (8 threads, one thread per CPU core) |

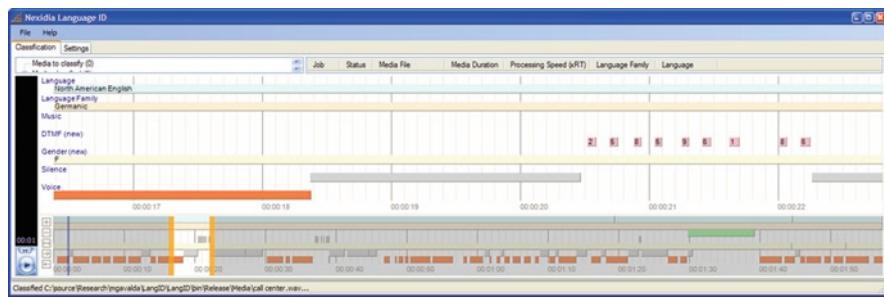
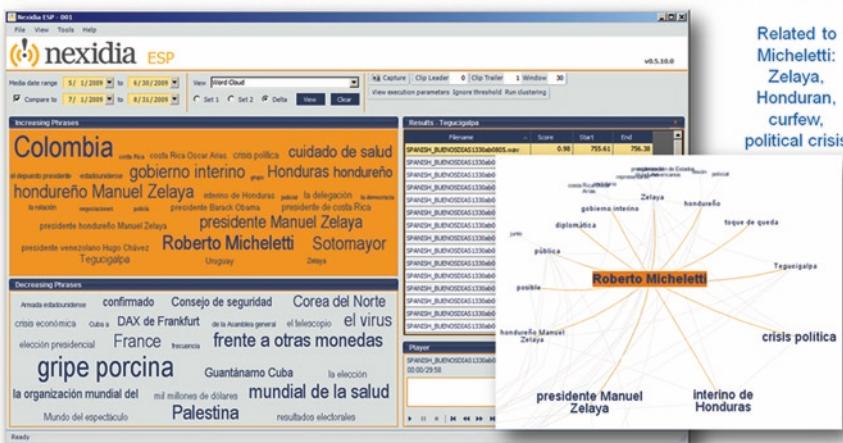


Fig. 10.12 Sample additional timelines extracted from the audio: language family, language ID, DTMF, silence, voice, music, gender, etc

Figure 10.12 shows an example of additional metadata tracks automatically generated for a call.

Top Movers: Colombia, Manuel Zelaya, Roberto Micheletti, Honduras, interim government; swine flue, WHO

Related to
Micheletti:
Zelaya,
Honduran,
curfew,
political crisis

Voice of America broadcasts, May-June vs. July-August, 2009

Fig. 10.13 Nexidia’s topic discovery solution applied to Spanish *Voice of America* broadcasts. The main window shows the top increasing phrases (top word cloud) and top decreasing phrases (bottom word cloud), while the inset (hypergraph) shows the phrases that tend to occur in close temporal proximity to “Roberto Micheletti”

10.3.6 Topic Discovery

One of the common criticisms of phonetic indexing and search is that it requires the end-user to know the terms and phrases that are subject to search, rather than having the system automatically identify these target phrases during the index process. However, there is no reason why searches cannot be automated as well. As a case in point, Nexidia’s ESP module provides topic discovery and trend analysis by automatically scanning textual sources for relevant search phrases, applying those search terms to the index, and presenting the results in compelling and informative visualizations (see Fig. 10.13).

As will be shown in the following section, this ability to automatically detect content, and understand both the frequency and relationship of spoken topics to one another, provides a method that dramatically enhances the early adoption of speech analytics in many market applications.

10.4 Current Applications of Speech Analytics in the Contact Center

Having defined the history of audio search and the development of speech analytics methods to improve the technology and bring it to market, this chapter now turns to the application of the technology to satisfy business requirements and provide a

return on investment both to its developers and its end-users. Within the last 5 years, speech analytics has achieved solid acceptance and penetration in a number of key markets, including:

- Intelligence and homeland security
- Internet and rich media
- Legal and regulatory discovery and review
- Contact centers

Because of the key role contact centers play in maintaining customer satisfaction and enhancing a company's business prospects, the information that is gleaned from effective use of speech analytics in this environment has enormous value for the market as a whole. For this reason, this chapter will focus on developments within the contact center environment.

10.4.1 Business Objectives in the Contact Center

At the most basic level, any contact center has two fundamental objectives:

1. Maintain or improve customer satisfaction and loyalty by handling customers' issues with quality and timeliness; and
2. Do all of the above with a constant eye on managing costs and efficiency.

With these objectives in mind, contact centers have deployed many different types of technology to route calls efficiently, to manage and improve agent performance, and even to provide self-service options to keep calls entirely out of the contact center. But call volume continues to rise, and speech analytics is becoming an essential element to understanding how to achieve the overall goals of customer satisfaction and cost containment.

10.4.2 The Speech Analytics Process Flow

Successful implementation of speech analytics in the contact center has a certain flow to it, not unlike the flow of information-to-action that is recognizable with any type of business intelligence. This flow is depicted in Fig. 10.14.

The deployment of speech analytics begins with initial discovery. Many companies new to speech analytics are not sure where to begin, and often ask the question "how do I even know where to look?" An automatic discovery process will mine through calls and identify those topics that occur frequently and those that are either growing or decreasing in importance. This initial narrative into calling activity identifies important issues that can form the foundation for a deeper analysis on key focus topics.

The real value in speech analytics is its ability to deliver quantitative intelligence from spoken audio information, and to turn this intelligence into actions that provide a meaningful return on the company's investment. This is demonstrated in



Fig. 10.14 Process flow for the application of speech analytics process in the contact center

steps B through F in the previous chart. This core process of speech analytics is aimed at accomplishing the following goals:

- B. Identify key customer behavior and call drivers and determine the magnitude of these call drivers on both cost and customer satisfaction;
- C. Identify the root causes behind these key call drivers, such as any business processes or environmental reasons for them;
- D. Determine the net economic impact to the company if these issues can be resolved;
- E. Develop an action plan to resolve them; and
- F. Continuously monitor progress and change to validate the business impact.

Once deployed in a fashion described above, the applications of speech analytics in contact centers can be varied, though they more or less have evolved into four main areas:

1. Streamlining business processes
2. Improving agent performance
3. Increasing market intelligence
4. Monitoring compliance

To better illustrate how speech analytics can be applied to each of these areas, the next section of this chapter will describe actual business use cases where companies have achieved tangible results with speech analytics in each area.

10.4.3 Streamlining Business Processes

One of the most-watched metrics in the modern contact center is that of first contact resolution (FCR). Improving the FCR rate - providing a satisfactory solution to a

customer problem within their first call - is a key contact center performance metric for two reasons:

- Each subsequent call increases the cost of providing that solution;
- Additional calls tend to lower customer satisfaction.

A variety of factors, including company policies and procedures or ineffective call-handling tools, may be limiting the agent's ability to close issues efficiently. But whatever the cause, effective use of speech analytics allows the contact center to easily identify the major factors that drive repeat calls, and implement steps to solve this problem.

10.4.3.1 Case Study in FCR

A major US wireless provider saw a spike in repeat calls related to one of its pre-paid phone products. Using speech analytics to drill deeper into this issue, they discovered that 15% of the calls were related to customers that had called after-hours to refill their prepaid card. However, these after-hours calls were routed to an outsourced contact center that did not have a direct link into the prepaid replenishment system and thus could not immediately apply the refill to the card. When customers tried to use their phones, they received an error message and then subsequently called back to inquire. This is a classic example of an internal company process that created both unhappy customers and increased costs. The simple solution - training the outsourced contact center to educate customers on when refill minutes would be available - was enacted immediately and helped eliminate the majority of repeat calls. As of this writing, the company is using the same speech analytics intelligence to determine whether or not to invest in direct integration between the outsourcer and its prepaid systems.

Another very important metric for managing contact centers is average handle time (AHT). A dramatic change in the average time it takes to handle each call can serve as an early warning of something unexpected or unusual taking place in the contact center. Temporary and predicted increases in AHT that are associated with new procedures, product releases, pricing changes and so on, can usually be offset by staffing adjustments or training sessions. But when AHT rises unexpectedly, and remains elevated, it is important to make adjustments quickly to minimize damage and any risk of negatively impacting customer satisfaction. As one can see, speech analytics provides an effective way to keep tabs on and improve AHT.

10.4.3.2 Case Study in AHT

A health insurance company in the United States noticed a problem with calls relating to one of its plan programs, whereupon the AHT for these calls was significantly higher than overall AHT for the organization. Applying speech analytics to the problem helped identify two very important aspects of this situation. First, using

a capability inherent within their speech analytics software that helped quantify “non-talk” time, they realized that a large percentage of this AHT was actually the accrued time that the customers spent being placed on hold. Second, speech analytics helped the health insurer to categorize the reasons that customers were put on hold, which helped very much to illuminate the root cause of the problem. Namely, these customers all had medical services performed outside their home “network,” and the call hold time arose when agents were engaged in the time-consuming task of contacting the other insurance carriers to try to work through such medical claims issues. Again, a business process was identified that consumed valuable network resources as well as significant agent and customer time. Modifying this process to handle claims issues offline resulted in more than \$600,000 in annual savings for just this one aspect of the contact center, in addition to providing a measurable improvement in customer satisfaction ratings.

10.4.4 Improving Agent Performance

It is no secret that contact centers spend a great deal of time and money training agents to be as effective and “customer friendly” as possible. And contact center agents are now asked to support customers across a range of issues that relate to overall company practice. But with turnover in the agent ranks reaching as much as 30% per year, contact centers need every advantage possible to make sure that agents can perform at their maximum potential.

One of the recent advances in speech analytics is the speed and accuracy with which these more advanced technologies can process audio content. And with these advances, these technologies can now be cost-effectively deployed as a real-time solution to help analyze content in contact center conversations, as they are occurring in real time, and help improve an agent’s ability to handle calls efficiently.

In a true real-time monitoring (RTM) application, speech analytics is tied into both the contact center’s switching network and its corporate knowledge base, with a specialized set of queries that are constantly monitoring for different combinations of words and phrases that relate to important topics that a customer may be addressing (see Fig. 10.15). Thus, when any given topic is spoken, the system will automatically retrieve the relevant content from the corporate knowledge base and present it to the agent so they can handle the issue during the call. This is a significant improvement over the current manual approach, during which an agent may have to navigate three or more levels deep in a database and still search for the best information before providing it to a customer.

10.4.4.1 Case Study in Real-time Monitoring

A telco in the Asia-Pacific region performed a pilot program using a real-time “consultant assist” application tied into their corporate knowledge base. The pilot

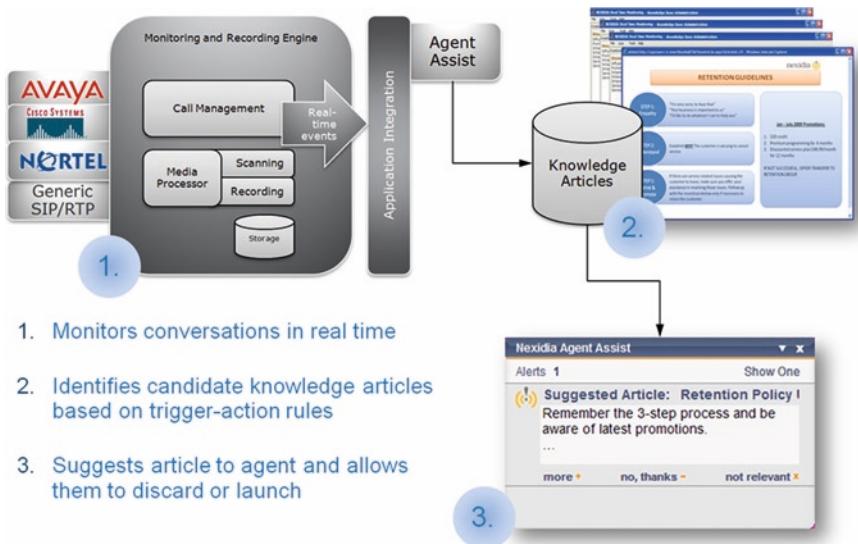


Fig. 10.15 Conceptual diagram of a monitoring system for assisting agents in real time. A phonetic-based engine taps into the switching fabric and can scan for thousands of audio streams. The recognition of phrases in a certain context triggers the presentation of knowledge articles and checklists to the agent and/or alerts to the supervisors with sub-second latencies

was a controlled experiment, where 60 agents were given the new application and 60 others were not. Both groups handled all incoming traffic as they normally would for a 90-day period. The results of this program were significant; the test group using the new RTM application saw an overall 8% reduction in their AHT, as the relevant information was delivered to agents much more quickly to handle customer issues. In addition, the test group showed a 2% increase in the revenue they generated through existing cross-sell and up-sell programs to customers, which was attributed to the fact that the system was more proactive in reminding them of these offers. All told, this pilot demonstrated a US\$2 million benefit to the contact center as a result of improved agent performance through the use of speech analytics in real time.

10.4.5 Increasing Market Intelligence

Companies spend millions of dollars annually for market research on their customers. These programs take many forms, from customer satisfaction surveys to product tests in focus groups, but their overall goal is the same: to provide market intelligence that companies can use to help make business decisions on products and other company processes.

Speech analytics is opening a whole new method by which companies can garner crucial market intelligence, and doing so with data that is both more quantifiable and lower in cost. Any company that has a contact center has access to

millions of hours of the “voice of the customer” and, with the appropriate speech analytics software, can mine this content to supplement, or even replace, its traditional market research tactics.

Speech analytics provides many benefits when compared to traditional market research:

- It is more quantitative and verifiable. In contrast to the random sampling method common in traditional research, speech analytics can be applied across 100% of recorded calls for a company to gather data from the broadest source possible.
- It is more timely, furnishing reports that can be delivered on a daily basis, or even multiple times per day, when compared to traditional research which may be delivered weeks after the data findings are collected.
- Data are gathered from actual customer interactions, rather than from a customer’s recollection of the interaction at a later date. As a result, the data are more likely to be in context and less subject to later “thought filtration” by either the customer or the data gatherer.

For all of these reasons, the data provided by speech analytics are becoming a critical component of how companies gather and act on their market intelligence.

10.4.5.1 Case Study in Market Intelligence

A contact center outsourcing company in the United Kingdom provides contact center capabilities to some of largest and most well known companies in Europe. As such, they must maintain a high standard of call quality and are driven by multiple key performance indicators (KPIs) from their clients. One of these KPIs relates to customer satisfaction and the level of positive and negative agent activity that customers express during a call. Whereas this information was previously derived from customer surveys done after the fact, they now collect these data using speech analytics across 100% of their recorded calls. As such, the data are available almost immediately, and they can quickly address issues that can help them maintain the high standards of service that their clients expect.

10.4.6 Monitoring Compliance

Contact centers are coming under increasing scrutiny and government regulations to maintain adequate standards of behavior and practice. Whether it is health insurance companies who must comply with the privacy restrictions in the Health Insurance Portability and Accountability Act (HIPAA), or collections departments that work under the auspices of the Fair Debt Collection Practices Act (FDCPA).¹

¹In 2008, the Federal Trade Commission received 78,838 FDCPA complaints, representing more than \$78 million in potential fines for improper collection activities (2009 FTC Annual Report on FDPCA Activity).

many contact centers are faced with the need to maintain strict compliance with expected practices or face consequences ranging from severe financial penalties to full criminal prosecution and loss of business. Speech analytics provides a cost-effective way to ensure that agents perform according to such expectations. This insures against potential liability and provides a mechanism to help train and improve agent performance in these challenging situations.

10.4.6.1 Case Study in Compliance Monitoring

A leading U.S.-based collection agency experienced a significant increase in financial penalties due to FDCPA violations and litigation. Management was concerned that continued violations would severely affect the company's long-term viability and profitability. Using speech analytics they quickly identified over \$200K dollars in potential FDCPA violations relating to agents' improper activity during the calls. By implementing appropriate training and monitoring across their collections center they have significantly reduced the agent activity that could lead to additional penalties.

10.5 Concluding Remarks

No longer an esoteric novelty, speech analytics are gaining acceptance in the marketplace as an indispensable tool to understand what's driving call volume and what factors are affecting agents' rate of performance in the contact center. The most advanced phonetic-based speech analytics solutions are those that are robust to noisy channel conditions and dialectal variations; those that can extract information beyond words and phrases (such as detecting segments of voice and music or identifying the language being spoken); and those that require no tuning (no need to create/maintain lexicons or language models). Such well-performing speech analytic programs offer unprecedented levels of accuracy, scale, ease of deployment, and an overall effectiveness in the mining of live and recorded calls. They also provide sophisticated analyses and reports (including visualizations of call category trends and correlations or statistical metrics such as ANOM on AHT by agent and call category), while preserving the ability at any time to drill down to individual calls and listen to the specific evidence that supports the particular categorization or data point in question, all of which allows for a deep and fact-based understanding of contact center dynamics.

Allowing for a gradual on-ramping process for the adoption of speech analytics solutions also helps in the marketplace. Forward-looking vendors typically offer a Proof of Concept as an initial validation of the technology, using the customer's audio, then a Quick Start as a hosted service to prove business value for a critical issue during a limited time, followed by an On Demand offering (hosted solution that provides trends and insights on an on-going basis), and finally a Licensed, on-premise deployment. Being able to integrate with a variety of recording platforms and telephony environments obviously widens the marketplace as well.

Looking ahead, we see speech analytics as a fundamental component in the cross-channel, real-time awareness fabric that will allow contact centers to feel and adapt to the pulse of their customers and the public in general.

References

1. Ch. Alberti, M. Bacchiani, A. Bezman, C. Chelba, A. Drofa, H. Liao, P. Moreno, T. Power, A. Sahuguet, M. Shugrina, O. Siohan, “An audio indexing system for election video material,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4873–4876, 2009.
2. M. Bacchiani, “Trends Toward Natural Speech,” Presentation at SpeechTEK, New York City, August 24–26, 2009.
3. P. Cardillo, M. Clements, M. Miller, “Phonetic searching vs. large vocabulary continuous speech recognition,” International Journal of Speech Technology, January, pp. 9–22, 2002.
4. M. Clements, P. Cardillo, M. Miller, “Phonetic Searching vs. LVCSR: How to Find What You Really Want in Audio Archives,” AVIOS, San Jose, CA, April 2001.
5. M. Clements, M. Gavalda, “Voice/audio information retrieval: minimizing the need for human ears,” in Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, Kyoto, Japan, December 2007.
6. J. Garofolo, C. Auzanne, E. Voorhees, “The TREC spoken document retrieval track: a success story,” Proceedings of TREC-8, Gaithersburg, MD, pp. 107–116, November 1999.
7. M. Gavalda, “Speech analytics: understanding and acting on customer intent and behaviour”, in presentation at the Business Systems Conference on Improving Performance in the Contact Centre, London, November 2009.
8. D. Graff, Z. Wu, R. McIntyre, M. Liberman, “The 1996 Broadcast News Speech and Language-Model Corpus,” in Proceedings of the 1997 DARPA Speech Recognition Workshop, Chantilly, VA, 1997.
9. D.A. James, S.J. Young, “A fast lattice-based approach to vocabulary independent wordspotting,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Adelais, SA, Australia, Vol. 1, pp. 377–380, 1994.
10. S.E. Johnson, P.C. Woodland, P. Jourlin, K. Spärk Jones, “Spoken document retrieval for TREC-8 at Cambridge University,” in Proceedings of TREC-8, Gaithersburg, MD, pp. 197–206, November 1999.
11. D. Jurafsky, J. Martin, *Speech and Language Processing*, Prentice-Hall, Upper Saddle River, NJ, 2000.
12. K. Ng, V. Zue, “Phonetic recognition for spoken document retrieval,” in Proceedings of ICASSP 98, Seattle, WA, 1998.
13. B. Ramabhadran, A. Sethy, J. Mamou, J.B. Kingsbury, U. Chaudhari. “Fast decoding for open vocabulary spoken term detection.” in *Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association For Computational Linguistics, Companion Volume: Short Papers* (Boulder, Colorado, May 31–June 5, 2009). Human Language Technology Conference. Association for Computational Linguistics, Morristown, NJ, pp. 277–280, 2009.
14. R.R. Sarukkai, D.H. Ballard, “Phonetic set indexing for fast lexical access,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, no. 1, pp. 78–82, January 1998.
15. J. Wilpon, L. Rabiner, L. Lee, E. Goldman, “Automatic recognition of keywords in unconstrained speech using hidden Markov models,” IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 38, no. 11, pp. 1870–1878, November 1990.
16. R. Wohlford, A. Smith, M. Sambur, “The enhancement of wordspotting techniques,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Denver, CO, Vol. 1, pp. 209–212, 1980.

Part III

Clinics

Chapter 11

Dr. “Multi-Task”: Using Speech to Build Up Electronic Medical Records While Caring for Patients

John Shagoury

Abstract This chapter discusses speech recognition’s (SR) proven ability to enhance the quality of patient care by increasing the speed of the medical documentation process. The author addresses the history of medical dictation and its evolution to SR, along with the vast technical improvements to speech technologies over the past 30 years. Using real-world examples, this work richly demonstrates how the use of SR technology directly affects improved productivity in hospitals, significant cost reductions, and overall quality improvements in the physician’s ability to deliver optimal healthcare. The chapter also recognizes that beyond the core application of speech technologies to hospitals and primary care practitioners, SR is a core tool within the diagnostics field of healthcare, with broad adoption levels within the radiology department. In presenting these findings, the author examines natural language processing and most excitingly, the next generation of SR. After reading this chapter, the reader will become familiar with the high price of traditional medical transcription vis-à-vis the benefits of incorporating SR as part of the everyday clinical documentation workflow.

Keywords Electronic health records (EHR) • Speech-enabled electronic medical records (EMR) systems • Medical transcription • Radiology reports • PACS • RIS • Front-end and background speech recognition • Natural language processing (NLP) • Record-keeping errors • Quality of patient care

11.1 Introduction

Since the first speech-driven clinical documentation and communication product became available in the U.S. during the late 1980s, speech recognition (SR) technology has improved the financial performance of healthcare provider organizations by

J. Shagoury (✉)

Executive Vice President of Healthcare & Imaging Division,
Nuance Communications, Inc., 1 Wayside Road, Burlington, MA 01803, USA
e-mail: Holly.Dewar@nuance.com

reducing the total cost of traditional medical transcription. It has enhanced the quality of patient care by making it possible for critical care physicians to immediately document and share information about patients in the emergency department, intensive care unit, or the surgical suite. SR also has enabled specialists to dictate, review, approve, and send the results of a clinical examination to a patient's primary care physician before the patient has even left his/her office.

SR is helping healthcare provider organizations to maximize the productivity of medical transcriptionist staffs, thus reducing an organization's reliance on traditional medical transcription and in some cases, eliminating the need for transcription entirely. The technology is helping physicians improve the thoroughness and consistency of their clinical documentation, navigate through electronic medical record (EMR) systems, and spend less time documenting patient care so they can spend more time delivering it. By changing the role of a medical transcriptionist from typist to more of an editor, SR is helping medical transcriptionists complete the actual work of transcription more quickly and effectively, thus reducing the effects of repetitive stress injury.

SR is now being used by more than 250,000 clinicians in more than 5,000 healthcare provider organizations to document patient care. The technology has been widely adopted by the innovators in healthcare delivery. As of the end of 2009, SR is being used to document patient encounters by 100% of the *U.S. News and World Report* Honor Roll Hospitals, 74% of the Most Wired Hospitals, and 73% of the Top 15 Connected Healthcare Facilities. SR is becoming a key adjunct for enabling the electronic transfer of clinical information in real-time, increasing the accuracy and consistency of transcription, and fostering the adoption and acceptance of the electronic health record (EHR).

11.2 Background

11.2.1 SR In and Outside of Healthcare

SR provides value for any industry that relies on dictation and transcription, including education, finance, government, insurance, and law. However, the technology is especially beneficial for healthcare because of the industry's enormous demand for dictation, transcription, and the need to cost-effectively manage resources.

Industries outside of healthcare do not generate nearly as much dictation. In healthcare, *every* patient encounter requires a documented report describing the reason for the visit, the patient's symptoms, physical findings, examination results, and recommendations for treatment and referral. A physician can encounter from 20 to 40 patients each day. A hospital with 1,000 physicians can handle tens of thousands of encounters as patients flow through the system. As part of the traditional clinical documentation workflow, a summary of each of these encounters must be dictated by the physician, transcribed by a medical transcriptionist, and returned to the physician for review and signature. No other industry is required to provide so much documentation.

Nor do other industries have such compelling need to streamline transcription turnaround time. If the final report of dictation that has been sent to a medical transcriptionist is delayed until an entire queue of dictations has been transcribed, decisions on patient care lie in wait. If reports are not processed quickly, healthcare provider organizations can fail requirements set by regulatory bodies and accrediting agencies. To meet standards of the Joint Commission on Accreditation of Healthcare Organizations, for example, hospitals must create written reports of all surgical procedures within 24 h of every operation.¹

Industries outside of healthcare follow different economic models, which allow them to pass along the cost of dictation and transcription to customers. In healthcare, however, the costs of dictation and transcription are oftentimes borne by physicians' offices and healthcare institutions, and the amount can be staggering. The American Medical Transcription Association estimates that as much as \$7–12 billion is spent every year to transcribe clinical dictation into medical text.² These costs are projected to rise steadily while the U.S. medical transcription market grows to \$16.8 billion in 2010.

11.2.2 *The Burden of Traditional Medical Transcription*

Of the total amount of medical dictation and transcription done in the U.S., roughly half is handled in-house by hospitals or physicians' offices. Most of the 98,000 medical transcriptionists, who were employed in 2006, the last year for which data are available, worked in hospitals or physicians' offices. Hospitals and physicians' offices assume not only the hourly wage for each transcriptionist, which ranges from \$10.22 to \$20.15 per hour,³ but also provide employee benefits, training, and other support, which can raise the total yearly cost for each staff transcriptionist to more than \$45,000.

An increasing percentage of the medical transcription business is being done by offsite transcription services in the U.S. or other countries. Although hospital-based medical transcriptionists performed the lion's share of dictation and transcription during the 1970s, only 53% of hospitals were using internal transcriptionists exclusively in 2003.⁴ The offshore transcription industry, which already accounts for about \$5 billion per year, is projected to increase to \$8.4 billion by 2010.

Whether done in-house or outsourced, medical transcription carries a heavy financial burden for individual healthcare institutions. As an example, each of the three

¹ Drum D. (1994). Medical transcriptionists feel the heat of hospital cost cutting efforts. *The Los Angeles Business Journal*, Feb 14.

² Atoji C. (2008). Speech recognition gaining ground in health care. *Digital HealthCare* July 22.

³ Bureau of Labor Statistics (2008–2009). Medical transcriptionists. *Occupational Outlook Handbook*.

⁴ Market Trends Inc (2002). Perceptions are reality: Marketing a medical transcription service. Survey data for Medical Transcription Industry Alliance.

hospitals that comprise Saint Barnabas Health Care System in New Jersey spends between \$374,854 and \$576,761 a year to outsource their medical transcription.⁵

11.2.3 The Cost Benefit of SR

SR has been a major means of reducing the cost of traditional medical transcription for healthcare facilities. In fact, more than 30% of the institutions that use one form of SR have saved more than \$1 million over a period of two or more years.⁶ The Camino Medical Group of Sunnyvale, CA, was able to cut \$2 million from its annual transcription expense by eliminating outsourced transcription.

Maine Medical Center, a 606-bed referral center, teaching hospital, and research center in Portland, saved more than \$1 million between 2002 and 2005 by improving the productivity of in-house transcriptionists, thereby decreasing the overall need to outsource medical transcription, hire temporary staff to cover for vacationing transcriptionists, and absorb additional overhead and recruitment costs for new hires.

The University of North Carolina (UNC) Health Care Hospitals, which serves more than 500,000 patients per year in its networks and clinics, realized \$1.169 million in cost savings between 2003 and 2006. The health system saved 70 cents for each of the 16.7 million lines of transcription it generated on average per year.

Physician practices also can benefit from SR cost savings. While a full EMR system is out of the economic question for most private physician office practices, SR is affordable - and it can cut transcription expenses between \$10,000 and \$30,000 per year.⁷ Most physician practice users see a return on their investment in SR within 3–12 months.

A key driver of cost reduction is increased productivity. A medical transcriptionist can complete three to four times more reports per hour using SR than by merely typing. A transcriptionist who types 50 words per minute can produce a 300-word document in 6 min. Even a highly accomplished transcriptionist who types 90–100 words per minute cannot compete with SR technology, which recognizes 150–160 words per minute at an accuracy rate up to 99%. With more than 300 physicians practicing in 10 branch clinics, outpatient centers, and the 300-bed Carle Foundation Hospital, Carle Clinic in Urbana, IL, is one of the largest private physician groups in the country. Soon after it adopted SR in 2004, Carle Clinic saw a productivity increase of 50% among in-house medical transcriptionists. By 2007 productivity was 100% greater as transcriptionists were able to edit twice as fast as they could type.

⁵ Forsman JA (2003). Cutting medical transcription costs. *HFM* July, p. 2.

⁶ These savings were realized by institutions that use eScription computer-assisted speech recognition from Nuance Communications, Inc.

⁷ Glenn TC (2005). Speech recognition technology for physicians. *Physician's News Digest*. May.

Greater productivity means quicker turnaround time. UNC decreased turnaround time from 14 to 17 h on average down to 4 or 5 h.

Along with quicker communication comes faster clinical decision making. In the emergency department, SR has been incorporated into wireless PDA devices so that on-the-scene physicians can record their findings on the move and other clinicians can gain access to the information before the end of a shift. SR is being used in conjunction with EMR systems to accelerate documentation by replacing hunt-and-peck keyboard-based population of data fields with voice-controlled navigation of templates and macros.

Despite the rapid growth and adoption of SR in healthcare settings, the technology has penetrated perhaps only 10–20% of the entire U.S. market. Spurred by cost savings and increased efficiency, as well as the ability to power the use of EMR systems, the market for SR is expected to double in size by 2013.⁸ SR is becoming critical to the mission of healthcare provider organizations and physician office practices as they seek to cut costs and improve the quality of patient care. The technology also is becoming critical to the widespread use and success of electronic medical reporting to improve both the efficiency and effectiveness of patient care.

11.2.4 From Dictation to SR

Research into SR technology began as far back as 1936 when AT&T’s Bell Labs, various universities, and the U.S. government worked independently on projects that would enable machines to respond to spoken commands and automatically produce “print-ready” dictation. The technology did not leave the laboratory, however, until the 1980s when James and Janet Baker founded Dragon Systems to make automated SR commercially available.⁹

Developments in SR technology quickly followed. In 1984, Dragon Systems released the first SR system for a personal computer that would open files and run programs through spoken commands. In 1986 the company began working on a continuous SR program with a large vocabulary, and in 1988 it launched a discreet SR system for personal computers that had a vocabulary of about 8,000 words. Two years later, Dragon Systems introduced the first speech-to-text system for general purpose dictation. Dragon NaturallySpeaking, a continuous speech and voice recognition system with a general purpose vocabulary of 23,000 words, as well as a continuous speech and voice recognition system for desktop and hand-held PDAs, followed in 1997.

SR for healthcare dates back to the work of Raymond Kurzweil, who founded Kurzweil Computer Products, Inc., in 1974 to develop a computer program that

⁸ Atoji, p. 1.

⁹ History of speech & voice recognition and transcription software. www.dragon-medical-transcription.com/history_speech_recognition.html.

could recognize text written in any font. Kurzweil Voice System technology powered the Kurzweil Clinical Reporter, a family of voice-activated clinical reporting systems for emergency medicine, triage reporting, diagnostic imaging and radiology, surgical and anatomical pathology, primary care, office-based orthopedic surgery, invasive cardiology, and general reporting.¹⁰

Early SR efforts in healthcare were slow and cumbersome, requiring dictating physicians to speak slowly and to pause between words and phrases to make sure the system could accurately recognize what they were saying. SR had to be “tuned” to be used by a particular speaker, and it had small vocabularies, or a complicated syntax.

Advancements in microchip technology and computing power have increased both the memory and the speed of operation of SR systems. As a result, the vocabularies of discrete words contained in SR programs have increased dramatically. Discrete vocabularies grew to 30,000 words in the 1990s and now include about 100,000 healthcare-specific active words that cover 80 clinical specialties.

Improved acoustic modeling and comprehensive statistical analyses are adding context to spoken words. Because of improvements in microprocessor technology, computer memory, and hard-drive power, SR algorithms can perform more analytical loops in shorter periods of time. When physicians dictate directly into their computers, words appear on the screens within half a second.

Refinements in acoustical and language modeling have increased the accuracy with which SR can recognize variations of the elements of the spoken word and select the most appropriate word for the clinical situation. As a result, SR systems for healthcare now readily recognize words, phrases, and even dialects so physicians no longer must speak haltingly; they can dictate at normal speeds and their sentences and paragraphs will be accurately reported 99% of the time.

Natural language processing (NLP) technology, which is still in its infancy, aims to add meaning so that spoken words are not simply transferred to text, but used to create “intelligent” narratives by automatically extracting the clinical data items from text reports and structuring them so they can be inserted in EMR repositories.

SR in healthcare has evolved to the point that it is replacing the keyboard. There are two major applications of SR in healthcare: (1) background SR, which improves the productivity of the traditional medical transcription workflow and (2) real-time SR, which allows for immediate documentation and reporting by specialists, such as radiologists, as well as primary care physicians and for speech-driven documentation to be entered directly into an EMR system.

With background SR, the dictation process does not change - the physician is still speaking into a tape recorder or digital device or personal computer after every patient encounter. The transcriptionist is still listening to what was recorded. Instead of typing out the individual words on a blank computer screen, however, the transcriptionist is editing a speech recognized document, reading the report that appears on the computer screen, correcting any discrepancies between the words

¹⁰Kurzweil speech recognition (www.speech.cs.cmu.edu/comp.speech/Section6/Recognition/kurzweil.html).

she sees and hears, and preparing a final document for the physician to review and approve. With front-end SR, the physician can dictate into a digital device, read the report after it passes through the SR engine, make any necessary corrections, and sign off immediately - all with a minimum of keystrokes and there is no need to rely on transcription support.

11.2.5 SR and Physicians

SR has been a technological resource for pathologists almost since the first clinical documentation and communication products were introduced. SR is a natural application for pathology departments because the nature of the reporting is heavily narrative, encompassing not only the detailed gross descriptions of the color, texture, weight, and size of a tissue specimen but microscopic analysis, diagnostic commentary and conclusions as well.

SR software plus self-editing tools allow pathologists to create and review clinical reports in one simple step within minutes of dictation while the physicians are still looking at microscopic slides. The software also can populate templates for reporting frequent and recurring findings. Pathologists can complete entire blocks of standard text reporting by using trigger words and voice commands, and they can activate fill-in capabilities using microphone buttons or voice directives.

SR's accuracy has reduced the number of reports that need correction at Department of Pathology and Laboratory Medicine at the Hospital of the University of Pennsylvania from 40% down to 2%. At the same time, the availability of templates has increased productivity by 20–30% while improving the precision of pathology reporting by prompting University of Pennsylvania pathologists to include all pertinent information.

SR also has an effective tool for radiologists. Perhaps as many as 40% of radiologists in the U.S. are using the technology to improve workflow and productivity, and also to increase the accuracy and consistency of reporting. The most recent applications in SR technology, which link with picture archiving and communication systems (PACS), and radiology information systems (RIS), make it possible for radiologists to maximize efficiency by completing diagnostic reporting and forwarding their conclusions to referring physicians in real-time.

With SR technology, hospitals can maintain a significant number of imaging studies with only a handful of radiologists. Cook Children's Health Care System, a pediatric hospital in Tarrant County, TX, can process about 135,000 imaging studies each year with only three full-time radiologists by eliminating the back-and-forth transcription approval process. The hospital also has decreased average turnaround time for radiology reports from 20 h down to 6 h and saved about \$9,000 per month.

In addition to pathology, radiology, and emergency medicine, SR vocabularies have been created specifically for 80 clinical specialties and subspecialties, including cardiology, internal and general medicine, mental health, oncology, orthopedics,

pediatrics, primary care, and speech therapy. Based upon analyses of millions of real-world medical reports, SR vocabularies include detailed information about the proper spellings and pronunciations of words. SR systems apply statistical models to determine how words fit together in sentences, paragraphs, and documents.

Because of its comprehensive lexicon, language and acoustic modeling, and robust SR engine, current medical SR software is less prone to error. Physicians using one of the specialty versions of SR are 20% less likely to make an error than they were with previous medical SR software. This translates into a savings of 15 s for each error that would have required review and correction, or a total of 20 min per day. By eliminating the extra time needed to find and rectify errors, physicians can spend more time with patients, adding from one to two patient visits per day.

Physicians can shift more of their time away from paperwork and into actual patient care by automating documentation through EMR systems. Adoption of EMR systems nevertheless is still extremely low. Thus far, only about 5–10% of healthcare institutions have adopted EMR systems. Even among the physicians who have access to EMRs, few are using all of the automated features, such as those involving data entry, because populating data fields using a keyboard and mouse slows them down. Studies indicate that physicians spend about 15 h a week documenting their encounters with patients. The average encounter takes three to four times as long to document in an EMR than it does to dictate.

Collaboration between SR and EMR systems is changing all that. SR is beginning to be used in conjunction with EMR systems so that physicians can navigate through clinical information to find and review lists of prescribed medications and test results with a single voice command. Physicians who switch to SR can reduce the time they spend documenting in an EMR by 50%. According to a 2007 study by KLAS, 76% of physicians who control data entry into an EMR system via speech reported faster turn-around time, which contributes to better patient care.¹¹

As a result, more than 100,000 clinicians have chosen to use SR provided by Dragon Medical systems to dictate directly into an EMR in the last four years alone. The U.S. Army is making SR software available to 10,000 of its physicians so that clinicians can avoid manually typing and mouse-clicking to document patient care, and improve their interaction with the Armed Forces Longitudinal Technology Application (AHLTA) - the Military Health System's own EMR system.

Using voice commands rather than a keyboard, physicians can more quickly conduct searches, complete forms, and make and respond to queries within an EMR. SR technology is allowing physicians to more easily populate software programs for assembling documentation, charting, entering orders, writing prescriptions and instructions for patients, managing patient records, and complying with regulations.

SR technology that immediately recognizes and allows electronic approval of dictation, drives template as well as text completion by voice command, and incorporates previously dictated reporting that can help physicians modernize not only the way they report information, but how they use it.

¹¹ KLAS 2007 (www.healthcomputing.com).

11.3 Front-end and Background SR in Healthcare

Early developments in SR technology led to the creation of real-time or “front-end” commercial speech-driven documentation and communication products for many dictation-heavy industries. Not until the 1990s were computing power and SR technology sophisticated enough to produce background or “back-end” SR for healthcare.

11.3.1 *Front-end Real-time SR*

Front-end or real-time SR is a one-step instant and interactive process that begins when physicians dictate and ends when they approve a speech-recognized document. The speech-recognition engine resides on a physician’s computer. So, rather than type his or her descriptions of anatomical features, or a computed tomography or magnetic resonance imaging scan, or the results of chemical staining of a tissue specimen, or a stress echocardiography test, the physician speaks directly into a digital recording device connected to a PC. In less than a second, while the physician watches, the speech-recognition engine transforms utterances into words and displays them on the screen. The physician can make corrections, sign off on a report, and send it anywhere in an integrated EMR, PACS or RIS - all in a single sitting.

While many different kinds of doctors use it, front-end SR is employed primarily by physicians who are under special pressure to turn their reporting times quickly around, and whose clinical vocabularies are relatively limited, such as pathologists, cardiologists, and radiologists (see the section on SR for diagnostics). Front-end SR also is used by emergency department physicians to speed exam and initial treatment findings to other caregivers.

Because of increasingly more powerful, linguistic and acoustical modeling, the continuous SR technology that underlies front-end SR can process massive numbers of patient records quickly and accurately. Front-end SR handles reporting for up to 100,000 visits to the emergency department of Miami Valley Hospital, one of three hospital members of Premier Health Partners in Dayton, OH. The technology is so accurate that physicians can dictate 30 or 40 charts and find only one or two errors.¹²

The technology accelerates reporting by simplifying the collection of information. Using voice commands, emergency department physicians at Miami Valley Hospital can call up macros or templates from the EMR system, and the speech recognizer will insert them into the dictation. The speech-recognition engine finds the proper macro regardless of the synonym a physician may use: “back template,” “back strain template,” “lower back template.” It also includes reminder cards so that physicians can instantly import certain categories of information, such as normal values, from physical examination reports. (See the section on SR and the EMR for more on the role of front-end SR.)

¹² Shepherd A (2009). Vive la voce. *For the Record* 21(14), p. 24.

Because the technology is available at the point-of-care, it allows physicians to record findings from examinations and tests while still fresh in their minds, or to dictate only a few facts, such as the patient's name and the history of the presenting complaint. These serve as reminders that enable them to create a more complete report at the end of the day.¹³

11.3.2 *Background SR*

Background SR is a process that occurs outside the sight of the physician. The background SR engine does not reside on the physician's PC; rather, it sits on a remote server that runs in batch mode to process all the dictation made by physicians in a particular healthcare provider organization or by physicians from multiple locales that subscribe to a SR application service provider.

As a result, background SR maintains the standard process of dictation: the physician sees a patient, dictates the specifics of the patient encounter into a recorder, and sends it off so a voice file can be processed into a draft document that is sent to a transcriptionist for editing. Before background SR, the transcriptionist would listen to a recording, type out the audible words, and place them in a predetermined format (see the illustration for traditional transcription process). With background SR, the transcriptionist reviews and edits a voice file that has been processed by a speech recognizer into a first-pass transcription complete with formatting (see the illustration for background speech-recognized transcription). The first time the physician sees the results of the dictation, it is in a final document, ready for his or her approval. The physician does not interact with the transcription process or have control over the output while dictating.

While other industries utilize real-time SR, background SR is used only in the healthcare industry. It is widely available for handling medical transcription from as many as 80 different types of clinical specialists with wide-ranging clinical vocabularies. While front-end SR technology is the mode by which SR accelerates decision making at the point of care, background speech technology is the means by which SR generates cost savings and operational efficiencies enterprise-wide. According to findings from the 2007 KLAS survey mentioned above, of the more than 300 healthcare professionals surveyed, 76% of the respondents reported that front-end SR speeded the dictation/reporting cycle. Background SR, in contrast, was associated primarily with productivity and decreased costs. Sixty-nine percent of the professionals in the survey identified productivity as a benefit of background SR, and 49% reported cost savings.¹⁴

¹³ Shepherd, p. 25.

¹⁴ Means C (2009). Adoption curve on the horizon for speech tools. Product Spotlight Speech Recognition, Jan.

More than 3,000 healthcare organizations currently rely on background SR to process approximately 1.8 billion lines of transcription per year. These organizations are seeing productivity increases up to 100% as background SR transforms medical transcriptionists into clinical document editors and doubles the speed with which they handle dictation. As they increase the amount of dictation they run through background SR, 85 or 90% of organizations achieve high rates of productivity on nearly all of their dictation volume.

11.3.3 Technologies Behind Front-end and Background SR

The first real-time, front-end SR products were driven by continuous SR technology that applied acoustical models to divide spoken text into phonemes, which are the smallest distinctive components of verbal language, and rearrange them into words. However, dictionaries cannot be developed simply by listing the individual phonemes that comprise a single word. Phonemes are actually acoustic realizations that depend upon the pronunciation of groups of sounds and the speed of articulation. In order to build an accurate acoustic model of a person’s voice, SR technology takes into account hundreds of thousands of voice imprints from users in order to capture the various ways that phonemes occur in natural speech.

Modeling also has to adjust to realistic acoustic environments, including the setting when an individual is speaking on a high-quality phone in a quiet room versus a speaker phone with noise in the background and accommodate a multitude of other contextual factors.

Linguistic models were incorporated in SR technology to reassemble the recognized words into statements or conversations that made sense in context. Based upon clinically specific vocabularies and individual speakers’ own patterns of speech, statistical models “learnt from previous mistakes” to determine the correct spelling of terms, distinguish between words that sounded alike, and identify the words that were more likely to be spoken by a speaker in a particular clinical specialty. For example, statistical modeling informed the continuous SR system when the speaker meant “write” [a report] versus the “right” [lung].

The technological breakthroughs that brought back-end SR to fruition had their roots in the 1990s when scientists began to move beyond straightforward decoding of the words that were spoken during a dynamic real-time interaction. If a speaker was not going to directly interact with the output of the speech recognizer and verbally format the structure of a document as he or she went along, scientists had to develop models that would automatically accomplish formatting in the absence of the speaker.

NLP, which was emerging toward the end of the 1990s, addressed some of the formatting issues. Nevertheless, brand-new formatting technologies were needed to account for punctuation, numerals, sections headings, etc.

To be efficient and spare computer time and power, language modeling had to limit the search for likely words among all possible word sounds. Task-dependent models

narrowed searches to the types of conditions a clinician treated and the type of work product for which the clinician was dictating. These models reduced the search space so acoustical modeling had a better chance of decoding the correct words.

One of the most important technologies to support background SR involves corrective training, which takes the medical transcript generated by a transcriptionist and learns from the corrections the transcriptionist makes directly to the draft document. An algorithm known as probabilistic text mapping (PTM) functions like a feedback loop to collect corrections over a large body of data and build a body of intelligence that can be tapped to anticipate and make corrections ahead of time. PTM is a form of language translational technology known as transformational modeling that has been designed specifically for automated medical transcription.

Tools also speed and simplify processing once dictation is in the hands of the transcriptionist. Background SR must be easy to use by transcriptionists. Even if a speech-recognized draft document was technically accurate, background SR would not achieve productivity gains and cost savings if medical transcriptionists could not edit it efficiently. SR scientists therefore observed medical transcriptionists at work and introduced post-speech-recognition processing that eliminated some of the bottlenecks in the editing function. Background SR post-processing accelerates the speed of dictation playback when the speaker pauses or hesitates. Such capabilities allow medical transcriptionists to control the speed of dictation playback without changing the pitch of the speech, and they customize playback speed to match the historical performance of the speaker and the medical transcriptionist.

Enhanced language modeling and customized post-processing capabilities in what is known as computer-aided medical transcription (CAMT) make it possible for medical transcriptionists to use certain keys to shortcut keystrokes while editing punctuation, operate an independent cursor to continue dictation playback while editing anywhere in the document, and apply the pause suppression function to eliminate long periods of silence during dictation.

Technological advancements learn the structure and style of each physician's documents and automatically correct mistakes in grammar and punctuation, handle rephrasing, add new medical terminology, and standardize formatting for section headings depending upon the type of document the physician is dictating.

11.3.4 Case Study Reports of Background SR

According to feedback from more than 100 users, background SR is improving the productivity of in-house medical transcriptionists, and in the process cutting costs, and accelerating document turnaround time.

The top 25 users of background SR designed for handling medical transcription on site by many types of physician specialists across the entire enterprise report that they:

- Have transferred 86% of their transcription volume from standard medical transcription processes to SR. In the process, they are eliminating the need to support

medical transcription services and attendant costs. For eight users, the change has led to combined annual cost savings of more than \$1.9 million in the first year.

- Are achieving rates of productivity that are 98% higher than the industry average. The average monthly transcription rate with background SR is 347 lines per hour; the industry average is 175 lines per hour. Experienced and skilled medical transcriptionists are reaching even higher levels of productivity with background SR – an average of 493 lines per hour.
- Are dramatically reducing clinical document turnaround time. The average time for completing medical reports declined from 42 to 20 h, a 53% improvement in turnaround time, in eight facilities.

11.3.5 Productivity Gains

Carolinas Medical Center-NorthEast, in Concord, NC, has been identified as one of the top 100 hospitals in the country according to measures of organizational leadership and improvement. When the hospital adopted background SR in 2004, it started small, enlisting only a handful of transcriptionists and physicians in the exercise. The number of transcriptionists who became speech editors quickly jumped from seven to 18, and the number of physician speakers grew from 88 to 319. At the present time, Carolinas Medical Center-NorthEast is using background SR to generate 6.5 million lines of transcription each year and produce 90% of its clinical documents.

Transcriptionists at the hospital are editing 300 lines per hour on average; top performers are editing 470 lines per hour. These figures far exceed the industry average of 150–200 lines per hour for standard medical transcription. In 1 month alone, the transcription team was able to accommodate 20% more lines of dictation than they generated only 6 months earlier.

11.3.6 Cost Reductions

Like many major academic medical institutions, the University of Wisconsin Medical Foundation, Madison, was seeing the costs of medical transcription rise with an increase in the volume of dictations that needed to be transcribed and the demand for quick transcription turnaround. The medical center was averaging 110,000 dictated minutes per month. Even with medical transcription outsourcing support and staff overtime, turnaround time was more than 72 h for some reports.

Three months after turning to background SR, the hospital was able to bring one-third of its 397 physicians across 35 locations onto the system and train 74 medical transcriptionists to use editing tools. Only 6 months after adopting background SR, the University of Wisconsin Medical Foundation was able to completely eliminate transcription outsourcing as in-house transcriptionists achieved productivity gains as high as 57%. By eliminating outsourcing, the medical foundation cut \$480,000 in annual expenditures for medical transcription.

Since Carle Clinic switched from traditional medical transcription to background SR in 2005, it has saved more than \$2 million in transcription costs. The clinic processes more than 47 million lines of transcription every year. Background SR technology is now used by more than 650 clinicians to complete dictation from clinical notes to correspondence to emergency department reports. The clinic has increased the capacity of its in-house transcription department from 13.3 million lines of transcription per year to 19 million lines and trimmed the size of the full-time transcription staff by four.

Beth Israel Deaconess Medical Center, Boston, MA, has saved more than \$5 million in transcription costs since it adopted background SR in 2002. The medical center, which serves nearly 250,000 patients each year, has been considered one of the most wired hospitals in the country. Yet, 7 years ago, only 40% of the clinical information collected by the medical center was recorded electronically, many of the reports on inpatients were hand-written, more than 400 different types of paper forms were used to record the process of care, and many of the documents in the patient chart were recorded only on paper.

Beth Israel Deaconess Medical Center currently produces more than 26 million lines of transcription by means of background SR. Nearly all (95%) of its total dictation volume is sent through the speech recognizer to prepare a first draft for editing. Transcription productivity nevertheless has at least doubled and in some cases tripled.

11.3.7 Turnaround Time Improvements

Health Alliance in Cincinnati, OH, a consortium of seven hospitals that serve the tri-state area of Ohio, Indiana, and Kentucky, employed as many as 110 transcriptionists in just one of its transcription departments and still had to obtain the services of outside transcription contractors to handle its clinical document dictation load in 2002.

After implementing background SR, the Health Alliance has seen document turnaround time drop 66%. Completed clinical reports now return to the physician within 10 h on average, 26–30 h faster than they were before. Background SR provides first-draft clinical documents, including procedure notes, discharge summaries, emergency department follow-up notes, and radiology reports, to more than 1,600 clinicians.

Medical transcription was one of the principal targets of a 2004 organization-wide program to eliminate inefficiency at Intermountain Healthcare, Salt Lake City, UT, an integrated healthcare delivery network that includes 21 hospitals and 150 clinics in the Northwest. Transcription services were highly fragmented; Intermountain had 42 different contracts with outside transcription firms for its clinics as well as numerous in-house transcription hubs. Transcription was not only inconsistent across the enterprise, it was tardy. Notes on operative procedures took more than 30 h to complete, and hospital discharge notes took 72 h.

Intermountain selected CAMT in 2006 so that it could streamline the number of document work types produced and share workloads across transcription groups.

In two years, the healthcare system was able to concentrate transcription workflow in one central in-house transcription department and reduce the number of external transcription services to three. It also decreased the number of document work types from 200 to 50. Intermountain reduced the overall document turnaround time by up to 83%. Turnaround time for operative notes dropped to 11 h, and the average turnaround time for discharge notes fell to 12 h.

11.4 SR in Diagnostics

When SR technology became robust enough to capture and record speakers while they were dictating in normal tones and cadences, it found a home in the diagnostic area of healthcare, particularly within radiology. SR technology has quickly become a natural tool for diagnostics. Unlike other areas of medicine, which have broad lexicons, the context and lexicon of diagnostics are comparatively limited. The ways in which radiologists document their findings involve a relatively confined set of words. Language models with SR dictionaries ranging from 50,000 to 70,000 words can accommodate the needs of radiologists. So, even in the early days of SR technology development, language modeling could reach high accuracy rates.

SR technology also met the demand of diagnosticians for speed and efficiency. Front-end (real-time) SR could be folded into the existing workflow processes in academic hospitals, community medical centers, and freestanding imaging centers. As a result, radiology reports could be generated in seconds while radiologists were still conducting a first-pass review of a patient’s diagnostic scans. As a result, radiologists did not have to spend extra time double-checking the reports they received from medical transcriptions days later. Before SR technology became available, radiologists at the Ottawa Hospital, the largest teaching hospital in Canada, were routinely waiting 10 days to receive diagnostic reports from medical transcriptionists. Especially during busy periods, radiologists would not have a report in hand for as long as 12–14 days. After adopting SR, Ottawa Hospital radiologists could release STAT reports to emergency department physicians within an hour.

Over the years, SR has incorporated additional voice-command tools that reduce the need for elongated direct dictation by radiologists. As an example, voice-driven macros allow radiologists to automatically insert into their imaging reports canned text that reflects normal findings, so they need not report individually on each of a patient’s organ systems that show no abnormalities on imaging studies.

With voice-activated templates, radiologists can populate bracketed fields within standard text segments of a radiology report simply by speaking into a microphone. Radiologists, therefore, can add details that reflect the specific aspects of a diagnostic case, including anatomic or procedural variables, or they can append additional dictation to standard blocks of text.

In some cases, natural language understanding and processing algorithms can help to automate report structuring. As an example, after the radiologist dictates statements about specific abnormal imaging scan results, applications logic within

some SR technologies can assign each of the findings to the appropriate data fields in the radiology report.

11.4.1 SR/IT Integration

SR technology can be easily linked with PACS and RIS by means of “Health Level 7” (HL7) data interfaces, which are used across the healthcare IT industry to transmit and receive radiology reports throughout a healthcare enterprise. Desktop integration makes it possible for SR technology to coexist and to operate in the background on PACS/RIS workstations. Thus, radiologists can immediately deliver diagnostic reports to any of the clinicians who are authorized members of a healthcare enterprise IT network. Radiologists also can take advantage of the shared HL7 data stream to consolidate order information and obtain appropriate images for review, simplify the process of analyzing data, perform trend analysis, measure exam productivity, launch customized data mining and analysis tools, and dictate radiology reports from any location – the radiology department or an offsite location, including the radiologist’s home. Because of the capability to access, review, and complete their diagnostic reports from any location, radiologists at the University of Southern California’s Keck School of Medicine can turn around radiology reports in less than 4 h.

11.4.1.1 Modes of SR Technology Deployment in the Radiology Department

SR technology in radiology is commonly utilized in real-time as radiologists dictate, review, and edit their own reports. So-called “once-and-done” SR allows radiologists to view speech-recognized text during or after dictation and edit the text by using a keyboard, mouse, conventional word processing tools, or by voice editing, which employs voice commands and microphone controls to correct and navigate from data field to data field within their documents.

Radiology Consultants of Iowa, Cedar Rapids, provides imaging interpretation services for two urban hospitals, seven rural hospitals, and an imaging center. Following a “once-and-done” approach, Radiology Consultants self-edits 97 percent of the nearly 1,000 diagnostic reports generated each day. Although the radiology practice has not directly correlated productivity with the implementation of SR technology, it saw an increase of 12% in reporting productivity within the first year after implementation and an increase of 28% in the first eight months of the next year. Radiology Consultants have also seen improved accuracy in reporting diagnostic findings. A total of nine errors were found in 493 speech-recognized reports compared with 13 errors in 283 traditionally transcribed reports. The error rate in SR-generated reports was 0.6 and 2.0% in traditionally transcribed reports.¹⁵

¹⁵ Radiology case study (2008). Radiological Society of North America case study report. Jan 8.

As an alternative, SR may be employed with delegated editing. With this option, radiologists do not edit their own diagnostic reports; rather, they send speech-recognized drafts to a medical transcriptionist/editor who corrects and formats the documents.

Little Company of Mary Hospitals in Torrance and San Pedro, CA, and Del Amo Diagnostic Center in Torrance apply both front-end and back-end SR technology in radiology. The hospitals and the imaging center together generate 145,000 diagnostic reports every year, a 60% increase in the volume of exam reports since they acquired SR technology. Overall turnaround time for radiology reporting nevertheless has declined from 32 to 8 h and turnaround time for STAT reports has decreased from hours to minutes.

Four of the 18 radiologists on staff at Little Company of Mary Hospitals and Del Amo Diagnostic Center self-edit all of their speech-recognized radiology documents, and 14 do at least some self-editing. Still, only 30–50% of the radiology reports are self-edited each month. The remaining speech-recognized radiology documents are sent to medical transcriptionists who edit the drafts. A workload that previously required 6.5 full-time medical transcriptionists is now being done by 2.5 staff members. Additionally, medical transcriptionists have increased monthly productivity levels by 30–50%, which translates into a saving of \$336,000 a year.

Radiologists can dictate and edit documents by speaking directly into a speech-driven radiology reporting solution.

11.5 SR and the EMR

EMRs date back to the 1960s when a physician named Lawrence L. Weed introduced the idea of computerizing medical records so the information they contained could be used more efficiently to both improve the delivery of patient care and reduce its cost. As part of his work with the University of Vermont, in 1967 Dr. Weed developed what is known as the problem-oriented medical record (POMR), which was intended not only to provide physicians with timely and sequential data about their patients but also to help obtain information for epidemiological investigations as well as clinical studies and business audits (see footnote 1).¹⁶

A POMR was used for the first time on a medical ward in 1970, when physicians entered data about the clinical histories, treatments, and results of their patients on a touch screen device. In the next few years, physicians were able to use the POMR to store information about patients, scrutinize their drug treatment, and serve as a safety check on the doses, potential interactions, allergic reactions, and side-effects of prescribed medications. More comprehensive EMR systems began to appear in the 1970s and 1980s.¹⁷

¹⁶Pinkerton K (2001). History of electronic medical records. Ezinearticles.com. <http://ezinearticles.com/?History-Of-Electronic-Medical-Records&id=254240>.

¹⁷Pinkerton (2001).

11.5.1 Benefits of EMR

University-based research centers and later private industries worked on EMR systems throughout the 1990s and into the 2000s because of their enormous potential effects on clinical data management. As alternatives to standard paper records, EMRs are far less cumbersome and labor-intensive to maintain. EMR systems eliminate the filing, retrieval, and refiling of paper records, the lack of access to files that have been checked out by a clinical department, or even the loss of critical patient information contained in records that have been misplaced. At least one estimate indicates that nearly 30% of paper records are not available during a patient's visit.¹⁸

EMR systems also replace the time-consuming and inefficient "hunt-and-peck" screening of paper records for analyzing, tracking, and charting clinical data and processes. As a result, EMR systems reduce the healthcare documentation load. According to a 2002 study in one medical center, clinicians using an EMR system took less than 90–135 min to prepare a discharge summary for a neonatal intensive care unit patient than to complete a paper report, and the EMR system saved medical-record professionals 4 min per patient record to chart, abstract, and code an uncomplicated case.¹⁹

Because they can quickly and readily be accessed, EMRs improve communication among healthcare professionals and thereby may decrease as much as 25–40% of the excessive cost to the U.S. healthcare system attributed to paperwork overhead and administration (see footnote 1). EMR systems also can improve the quality of patient care by providing decision support at the point of care. A computerized physician order entry (CPOE) system in and of itself could prevent 200,000 adverse drug events and save hospitals \$1 billion a year. In the ambulatory setting, a CPOE could avoid two-thirds of preventable adverse drug events and save \$1,000–2,000 per case.²⁰

11.6 EMR Adoption Rates

As the costs of healthcare delivery in the U.S. steadily rises, EMR systems have become an increasingly important target of investment by the federal government. Believing that computerizing medical information was "one of the most important things we can do to improve the quality of health and at the same time make the cost of health care more affordable," U.S. Health and Human Services Secretary Tommy Thompson in 2004

¹⁸ Expert System Applications, Inc. (2005). Saving using EMR vs. manual methods.

¹⁹ Arthur D. Little (2001).

²⁰ Hillstead R (2005). Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Health Affairs* 24(5) 1103–1117.

outlined a Bush Administration plan to create a nationwide system of EMR systems and encourage hospitals and ambulatory clinics to acquire information technology that could help save the U.S. healthcare system at least \$140 billion a year.²¹

Hoping to provide healthcare provider organizations with further incentives to computerize medical records, Congress set aside \$19 billion in 2009 to promote investment in EMR systems as part of the \$787 billion American Recovery and Reinvestment Act (ARRA). The Health Information Technology for Economic and Clinical Health Act (HITECH) portion of ARRA sets aside \$17 billion in direct incentive payments for physicians and hospitals that participate in Medicare and Medicaid programs to adopt or use EMR systems and \$2 billion in grants and loans to advance health information technology.²²

EMR adoption rates among healthcare provider organizations nevertheless remain low. Four years after Bush Administration efforts to promote a nationwide EMR system, most hospitals still lacked essential electronic reporting tools. The EMR Adoption Model (EMRAM) of the Health Information Management Systems Society (HIMSS) reported in 2009 that only 6% of hospitals had advanced EMR capabilities, such as computerized practitioner order entry, physician documentation, medical information warehousing and data mining, and full radiology PACS. Along a seven-stage path toward a fully paperless EMR environment, 31% of hospitals were at stage 2, meaning they had an electronic data repository of laboratory, pharmacy, and radiology data that could be reviewed by clinicians. Thirty-five percent of hospitals were at stage 3, meaning they had computerized nursing documentation and clinical support for nursing.²³

A study of 3,000 hospitals published in 2009 by the *New England Journal of Medicine* concluded that only 1.5% of all healthcare provider organizations had a comprehensive EMR system and only 8% had an EMR system installed in at least one unit.²⁴ It has been widely reported that only about 20% of 900,000 clinicians nationwide are currently using EMR software.

11.6.1 Barriers to EMR Adoption

The upfront cost of an EMR system has been a major deterrent for physician office practices. A drag on productivity is, however, another major disincentive. Several studies have demonstrated that physician productivity can decline by as much as 10% during

²¹ Bush Administration report recommends implementation of EMRs, other health care IT (2004). *Kaiser Daily Health Report*, July 21, p. 1.

²² Understanding ARRA EMR incentives and ROI (2009). ProTech Networks, p. 1.

²³ HIMSS (2009). Most U.S. hospitals within two steps of having essential EMR tools in place. Apr 14, p. 1.

²⁴ Chickowski E (2009). Speech recognition may speed EMR adoption. *Smarter Technology*, Aug 28, p. 1.

the first months after implementation of an EMR, and that, loss of productivity can mean an average drop in revenue of \$7,500 per physician.²⁵ One of the most frequent impediments to full EMR system implementation in the hospital setting is physician dissatisfaction with the changes they must make in their documentation workflow.

Physicians are reluctant to use an EMR system because it slows them down and prevents them from accurately depicting the patient encounter. Documenting a typical visit to a physician office into an EMR system can take three to five times longer than traditional dictation.²⁶ Physicians often feel that the EMR interferes with the stream-of-consciousness reporting they are used to and may cause them to omit important patient care details.²⁷

EMR systems were designed to provide a vital element to patient care data – structure – so that medical information could be codified for analysis. Structured data allows software to intelligently support patient care by including tools to help clinicians improve the quality of medical care and the efficiency of the practice of medicine. Reminder systems, for example, inform clinicians when a patient is due for follow-up or preventive care. Alerting systems flag contraindications among prescribed medications. Coding systems identify the correct billing codes for reimbursement.

To provide structure to patient care data, EMR systems are based upon point-and-click templates that require clinicians to check boxes, radio buttons, or choose from a pull-down menu to ensure that essential items of data have been captured and stored in a structured database format. However, templates cannot cover every patient situation. Nor can templates capture the underlying meaning and inference that can be found in contextual information or the relevance of and relationships between subjective observations. The loss of these forms of information can negatively affect patient care. A study conducted by clinicians and researchers using data from the Veterans Health Administration (VHA) EMR system showed that many adverse drug events occurred because the information captured electronically was incomplete and it was not stored directly in the EMR system (see footnote 1).²⁸

11.7 The Need for an Electronic Patient Narrative

By their very design, EMR systems threaten the traditional clinical narrative, which is the first-person “story” that is created by a clinician to describe a specific clinical event or situation. The clinical narrative allows clinicians to explain and illustrate

²⁵Sharma J (2008). The costs and benefits of health IT in cancer care. *Oncology Outlook*, Aug 21, p. 3.

²⁶Catuogno G (2007). The role and relevance of medical transcription to EMR adoption. *Executive HM*, p. 2.

²⁷Nuance Communications, Inc. (2006). Speech recognition: accelerating the adoption of electronic medical records, p. 2.

²⁸Hurdle J (2003). Critical gaps in the world’s largest electronic medical record: ad hoc nursing narratives and invisible adverse drug events. *AMIA Ann Symp Proc*: 309–317.

clinical practice and patient care decisions, so they can be shared and discussed with colleagues and used to guide future decision making.

To preserve the clinical, narrative and still structure patient care data, many healthcare provider organizations are embracing the concept of the electronic patient narrative, which permits clinicians to complete their clinical documentation in free text in their own words and enter the information into the EMR system. These organizations are making electronic patient narratives part of the standard practice of documentation by developing guidelines and teaching clinicians how to use the EMR system to insert narrative elements in their patient encounter notes. Organizations also are adopting SR technology, which helps physicians not only to record their observations but to populate EMR data fields.

SR technology allows physicians to maintain their traditional form of communication about patient encounters – direct dictation. Dictation is still the most popular form of documentation for physicians. Dictated and transcribed documents comprise about 60 percent of all clinical notes generated in the U.S. every year.²⁹ According to some EMR system vendors, up to 80% of physicians prefer direct dictation over direct data entry (via keyboard) into electronic systems.³⁰

SR technology supports two forms of electronic documentation:

- Speech-assisted transcription, in which a clinician’s dictation is captured, the voice file is sent and “recognized” by a speech-recognition engine as a first-pass step. A speech-recognized draft is created, reviewed, and edited by a medical transcription editor and released for review and signature by the clinician within the EMR. This type of back-end SR is 50% less costly than traditional transcription of medical records.
- Speech-driven or speech-enabled EMR systems, in which clinicians dictate directly into free-text fields of an EMR, review their dictations directly on the computer screen in real time, and edit as needed. This form of front-end SR technology is faster and less costly than traditional approaches to documentation creation. Traditional approaches include manually driven EMR, which requires clinicians to type their free-text narratives into an EMR system. Standard transcription, as discussed previously, requires physicians to dictate into a digital microphone or telephone, then wait for a medical transcriptionist to prepare a typed report of the dictation from scratch before he or she can review and approve the notes for entry into an EMR. Manually driven EMR does not incur the cost of medical transcription; however, it significantly interferes with a physician’s clinical productivity. Traditional transcription is the most labor intensive and least cost-effective method of documenting findings in an EMR.

²⁹ Association for Healthcare Documentation Integrity (2009). Medical transcription as a faster bridge to HER adoption, May, p. 2.

³⁰The role and relevance of medical transcription to EMR adoption, p. 3.

11.8 Natural Language Processing

NLP refers to the discipline in Artificial Intelligence that is focused on building systems that could mimic human ability to grasp meaning from spoken or written language. While the field of NLP is very complex and broad in scope, the focus in medical informatics is to automate the process of converting a clinician's narrative dictation into structured clinical data.

The vision of NLP provides clinicians the best of both worlds – allowing physicians to document care comprehensively, capturing the uniqueness of each patient encounter, unencumbered by the limitation of the rigid structure of documentation “templates” in their own words, yet have many key medical terms and findings identified within the dictated narrative and automatically saved in the appropriate fields of the EMR’s patient database, where information could later be analyzed, reported and used to produce actionable items.

NLP software analyzes “free text” dictation, tagging data elements that fall in the major categories of information needed by physicians such as clinical problems, social habits, medications, allergies, and clinical procedures. The tagged elements can then be used to populate the data fields within EMRs systems, enabling both retrospective analyses across large amounts of medical records, as well as medical decision support at the point of care for individual patients, to name just a few potential uses of structured data.

Some forms of NLP are being designed to tag and store sections of transcribed reports – such as History of Present Illness, Findings, Assessment, and Plan – so they can be individually accessed, reviewed and used at a later time. For patients with chronic conditions, such as diabetes, or those patients undergoing lengthy episodes of care, a feature known as “auto-reuse” obtains information from previous reports that normally would have to be re-dictated and re-enters that information automatically into a new document. Progress notes are one common form of clinical documentation which benefit significantly from auto-reuse. Early studies indicate that the auto-reuse functionality and SR-based templates can automate 50% or more of a variety of clinical reports.

In the near future, sophisticated NLP-powered SR technology may free physicians from having to manipulate complex pick-lists, drag-and-drop and keyboard functions, as well as filling out numerous fields on a screen, to complete EMR system templates. These advanced algorithms will automatically analyze the free narrative and extract the required information to complete the documentation accurately and comprehensively, enabling physicians to focus on patient care without having to worry about the mechanics of documenting each encounter.

11.8.1 *Advantages of SR-Powered EMR Data Entry*

SR helps physicians use EMR systems without changing their documentation routines. Physicians can dictate narratives in their own words; they can also enter any section

of an EMR system and dictate their comments and observations directly into the designated field. Physicians can use voice commands to move from one section of the electronic patient record to another.

As a result, physician’s productivity is neither affected nor increased. As mentioned earlier, the 2007 report by KLAS found that 76% of the clinicians using desktop SR to directly control an EMR system with speech could complete medical reporting more quickly than typing or using mouse navigation alone; 13% stated that they were more productive. Physicians could more easily conduct data searches and queries, write prescriptions, record aftercare instructions, and enter orders by dictating into an SR-driven EMR system than by keyboarding. Additionally, they could accelerate clinical decision-making by completing documentation and reporting exam and test results more quickly, and they could decrease the time they spent in documentation.

SR-driven EMR systems help improve cash flow and revenue. Because the technology enables patient notes to be completed almost immediately, hospital case workers and discharge planners can more quickly arrange for post-hospital care, so that patients spend less time in the hospital, and hospitals are not financially penalized for extra days of care while patients await post-hospital-care placement. Outpatient care centers can deliver charge capture information (procedure and diagnostic codes) and provide supporting clinical information for billing to insurers in a matter of minutes and therefore streamline the billing and collection process. Physician office practices can customize voice macros and templates to comply with billing guidelines, thereby increasing the accuracy and speed of billing.

11.8.1.1 A Case Study in Using an SR-Driven EMR System

Slocum-Dickson Medical Group is a multispecialty physician-owned clinical practice in New Hartford, NY, that has 75 physicians and 500 clinical professionals on staff. Slocum-Dickson decided to implement an EMR system in 2000 to support its vision of “one patient, one record, one system, and one schedule” so that patient notes would be completed, follow-up care would be scheduled, prescriptions would be written and sent to the pharmacy, and billing information would be ready for submission by the time they left their examination rooms.

Six years later, however, many physicians were still relying heavily on traditional medical dictation and transcription. Physicians were using the EMR system to document only a small portion of the patient encounter, such as clinical problems and medications. A rising volume of medical transcription increased the document turnaround times and administrative costs. At times, transcription turnaround times averaged 48–72 h, and physicians were spending \$12,000 a year on transcription. Also, due to the nature of transcription, some physicians had backlogs of up to 100 unsigned charts.

After only one day of training, most physicians in the medical practice were able to use SR to generate about half of their patient notes. Within a few days, most were dictating all of their medical decision-making notes into the EMR system, including

history and physical examination findings, assessments, and patient care plans. Physicians could use SR to add diagnostic detail to the descriptive history of a patient's present illness, increase the accuracy of their reporting by viewing documentation in real time, produce comprehensive referral letters for clinical specialists on the day they see a patient, and include supporting documentation for billing and reimbursement. SR technology saved physicians so much time that they were able to see one to two more patients a day, and it saved the group practice \$750,000 a year in medical transcription costs.

11.9 Perspectives on the Future

Major software companies already are predicting that SR will represent the “new touch” in computing. Speech-recognized and -operated computers are a “natural evolution from keyboards and touch screens,” according to the general manager of Speech Recognition at Microsoft Corp. “Speech is becoming an expected part of our everyday experience across a variety of devices,” including automobiles, smartphones, and personal productivity software with voice-activated navigation and search features, said Microsoft’s Zig Serafin.³¹

Of all the industries that can benefit from SR, healthcare perhaps tops the list. In fact, SR in healthcare presents a tremendous growth opportunity. While the majority of healthcare provider organizations already have some form of SR in use, there is ample room for additional deployments across new and existing departments, as well as for new applications of SR to enhance healthcare workflow. Documentation will remain critical and core to the patient care process and SR is the fastest way to transport dictated information from the human mind into a sharable, manageable and actionable form. SR, therefore, will likely grow at a continuous rate, resulting not only in significant gains in productivity and cost effectiveness for the traditional medical transcription process, but capturing information quickly and accurately for EMR systems, PACS, RIS, CPOE, and other electronic information systems. SR also will help the healthcare industry use EMR systems at maximum effectiveness in order to ensure that patients’ medical notes are robust and contain detailed information and are not simply point-and-click documents.

In the hospital setting, SR is proving that it can help reduce record-keeping errors and improve the overall quality of patient care. With the cost to transcribe physician dictation running between \$7 and \$10 billion per year, SR will continue to have a profound impact on hospital cost structures. Money that was previously spent on manual documentation processes can therefore be repurposed to patient care initiatives. SR also can reduce cross-infection by eliminating the need for sharing

³¹ Microsoft (2009) Spread the word: Speech recognition is the “new touch” in computing. Microsoft PressPass, October 28.

keyboards and allowing physicians to use their hands to tend to patients rather than operate a touch-screen device. It is no wonder, then, that some industry analysts believe every hospital will have SR infrastructure capable of supporting EMR systems within 5–10 years.³²

11.9.1 Next Generation SR

Given that in recent years much progress has been made in SR, with the bulk of the technology already developed but waiting to be applied, in the foreseeable future, SR will be able to offer “talk forms” that fill in the blanks in EMR-structured data fields.³³ The next generation of SR will not only recognize what a physician is saying, but may understand the narrative, identify specific data elements and metrics, and populate them in the appropriate structured data field. When a physician says, “BP 180 over 72,” SR will carve out the information as a blood pressure metric and automatically insert it in the physical examination field of the patient’s EHR. When a surgeon completes an operative report, SR will tease out the pieces of data that belong in the patient’s past medical history, the suture size, the operative course, and the postoperative medications. Neither physician will need to use a mouse or a keyboard to select the correct data field, or even use SR to navigate through the EHR and note the required clinical details. SR will act intelligently to scour the patient narrative for relevant bits of information and transfer them instantly to the proper location within the EHR.

NLP and its ability to intelligently process text and extract information will allow healthcare provider organizations to better understand the relationship between treatment pathways and patient outcomes while spotting and tracking trends in patient care. NLP will reduce the need for manually analyzing copious amounts of electronically captured data and information.

SR will also move toward decision support that will provide immediate feedback to physicians at the point of dictation, whether they are using a digital recorder, PDA, or mobile phone. SR is expected to become a leading means for capturing information into all healthcare information systems, including mobile devices. It should quickly find its way into healthcare-specific mobile applications that healthcare providers can use to document at the point of care and patients can use to quickly input their healthcare information both at the doctor’s office and from home.

³² Staygolinks (2009). Hospitals lead in speech recognition infrastructure, Oct. 27.

³³ Staygolinks.

11.9.2 The Challenges

Although SR currently achieves accuracy rates of 98–99%, core underlying SR technology will have to make NLP as close to perfect as possible to assure physicians that the data they dictate in the patient narrative will get to the right place in the structured EHR. It would serve no purpose if physicians have to recheck each data field that had been completed by SR. Consequently, SR will need to examine how medical information is codified by capturing vast amounts of data, assigning meaning to items of data, defining elements of data as “diagnoses,” “medications,” “physical findings,” etc., testing linguistic algorithms in real-life settings, measuring the results of testing, and continually refining the process.

Currently, SR software is sold with a headset or a microphone that offers the acoustic quality the speech-recognition engine needs to capture spoken words accurately. Recording capabilities in mobile devices do not have that level of acoustic quality (some mobile devices do have that capability, but not yet the majority of them). SR software companies will therefore need to work with mobile device manufacturers to ensure that their microphones are precise enough to produce high-quality sound even in the noisy environment of a busy hospital emergency department.

11.9.3 The Possibilities

How might SR be used in healthcare during the coming decades? Some applications appear to be logical, next-step extensions of present-day hardware and software. For example, SR may someday be incorporated in advanced technology that allows patients to voice-enter changes in their medical condition into their medical records so that physicians would have access to the most up-to-date information prior to the next office visit. Pilot projects already are underway that test programs (such as MyChart) linking home-monitoring hardware and documentation software. The hardware regularly takes standard measurements such as weight, blood pressure and glucose level, for patients with chronic diseases, including high blood pressure and diabetes. The software immediately enters the information into the patient’s medical record and alerts the physician whenever measurements are abnormal.³⁴ An SR capability would permit patients to add comments, descriptions, or explanations that amplify test results.

Other potential uses are within the realm of imagination. SR already leads computers to take specific steps. By means of voice commands, physicians can verbally ask their PC to search the internet for up-to-date information about a specific medication they wish to prescribe, and the computer will display information about

³⁴ Adler J and Interlandi J (2009) The hospital that could cure health care. *Newsweek*, December 7, p. 54.

the drug’s contraindications and potential adverse reactions. Could speech drive computers and other forms of electronic equipment take more complex types of actions? Could SR be used by surgeons to guide the movements of a surgical robot or power a motorized wheelchair for a paraplegic?

As computing power increases, the links between hardware and software naturally become more and more seamless; and as linguistic modeling gains precision and specificity, SR will move beyond transforming the way that physicians and patients create, share, and use documents to help merge information gathering with clinical practice. How that merger fully plays out remains to be seen.

Image of a structured medical record vs. an unstructured medical record. Structured medical records allow for uniform, easily accessible medical information and terminology. Medical transcriptionists and/or doctors can structure their notes manually or natural language processing (NLP) capabilities can automate structuring.

Image of traditional background speech recognition (SR) workflow, enabling clinicians to create documents in the most efficient way possible – by speaking into a phone, dictation device or electronic medical record (EMR), while background SR technology creates a high quality first draft that MTs quickly review and edit, typically doubling productivity when compared to traditional transcription.

Radiologist uses real-time speech recognition to document medical reporting. As he speaks, text appears on the screen for review and finalization of the document. No medical transcriptionist support is needed in this workflow.

Chapter 12

“Hands Free”: Adapting the Task-Technology-Fit Model and Smart Data to Validate End-User Acceptance of the Voice Activated Medical Tracking Application (VAMTA) in the United States Military

James A. Rodger and James A. George

Abstract Our extensive work on validating user acceptance of a Voice Activated Medical Tracking Applications (VAMTA) in the military medical environment was broken into two phases. First, we developed a valid instrument for obtaining user evaluations of VAMTA by conducting a pilot (2004) to study the voice-activated application with medical end-users aboard U.S. Navy ships, using this phase of the study to establish face validity. Second, we conducted an in-depth study (2009) to measure the adaptation of users to a voice activated medical tracking system in preventive healthcare in the U.S. Navy. In the latter, we adapted a task-technology-fit (TTF) model (from a smart data strategy) to VAMTA, demonstrating that the perceptions of end-users can be measured and, furthermore, that an evaluation of the system from a conceptual viewpoint can be sufficiently documented. We report both on the pilot and the in-depth study in this chapter.

The survey results from the in-depth study were analyzed using the Statistical Package for the Social Sciences (SPSS) data analysis tool to determine whether TTF, along with individual characteristics, will have an impact on user evaluations of VAMTA. In conducting this in-depth study we modified the original TTF model to allow adequate domain coverage of patient care applications.

This study provides the underpinnings for a subsequent, higher level study of nationwide medical personnel. Follow-on studies will be conducted to investigate performance and user perceptions of VAMTA under *actual* medical field conditions.

Keywords Voice-activated medical tracking system • Task-technology-fit (TTF) model • Smart data strategy • Medical encounter • Military medical environment • Shipboard environmental survey

J.A. Rodger(✉)

Professor, Department of Management Information System and Decision Sciences,
Indiana University of Pennsylvania, Eberly College of Business & Information Technology,
644 Pratt Drive, Indiana, PA 15705, USA
e-mail: jrodger@iup.edu

12.1 Introduction

The contents of this chapter are the results of an almost decade long odyssey that began in early 2002, relying on a government sanctioned grant that studied the impacts of gender on voice recognition. The results of such studies of gender and voice recognition were reported in a number of peer reviewed publications, such as the *International Journal of Human Computer Studies and Decision Support Systems* [42,43]. The original paper which became the basis of this journal article was presented at the *Decision Sciences Institute Conference* in San Francisco, California (2005). In 2009, 33 subjects were involved with the task technology fit (TTF) survey to measure end-user perceptions of VAMTA's technology acceptance from a *smart data* strategy.

The original studies of gender and voice recognition followed the Institutional Review Board process, for protecting the rights of human subjects, set forth by the Navy, which did not require that specific preconditions be met to conduct those studies. Such studies were government funded and the report was supported by the Bureau of Medicine and Surgery, Washington, DC.¹ The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, or the U.S. Government. The research was approved for public release, with unlimited distribution. Human subjects participated in this study after giving their free and informed consent.

The present study on user acceptance of VAMTA was conducted, likewise, in compliance with all applicable federal regulations governing the Protection of Human Subjects in Research. The informed consent form addressed five critical points:

1. Subject participation in the study was voluntary
2. A statement of the subject's right to withdraw at any time and a clear description of the procedures for withdrawal from the study without penalty
3. Subjects were informed of the level of risk ('no known risk') and the means of protecting the subjects from known risks or minimizing the risk
4. Confidentiality was ensured
5. The means by which confidentiality was ensured was elucidated.

These five points listed above were critical elements of the investigation. Thus, it was important to include enough specific and detailed information regarding the purpose and nature of our study to ensure that the study subjects were fully informed. A copy of the Informed Consent Form was given to each subject who participated in the study.

The VAMTA study had evolved from an initial feasibility study for testing the concept to validation of end-user perceptions of the acceptance of this technology. Not surprisingly, the literature reflects a similar pattern of evolution of the state of the art of speech recognition adaptation by end-users. The original feasibility literature, circa 2000, has evolved to the point of the actual reporting of VAMTA

¹ Work Unit No. 0604771N-60001.

findings themselves [43]. The updated literature review represents the 2010 reporting of the end-user perceptions of VAMTA task–technology fit and the smart-data strategy for optimization of performance. While this chapter integrates earlier works and new material, we agree with Scharenborg [44] and others that a decade ago researchers found many of the same limitations in speech recognition systems which still persist today. Nevertheless, it is the validation of the end-user technology acceptance model and the acceptance of VAMTA – as a fit between task and technology – that gives a more sanguine picture which places voice recognition on the cutting edge of smart data applications in healthcare.

Moreover, it cannot be denied that end-user acceptance will continue to be necessary in order to extend current research, such as integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition (ASR) [45] and in single channel speech separation [12]. And certainly as the medical community continues its quest for more efficient and effective methods of gathering patient data and meeting patient needs, one can naturally expect increased demands to be placed on information technology (IT) to facilitate this process. What is more, the situation is further complicated by the segmented nature of healthcare data systems.

Given that healthcare information is often encapsulated in incompatible systems with uncoordinated definitions of formats and terms, it is essential that the different parts of this organization, which have different data systems, find ways to work together in order to improve quality performance. VAMTA was developed partly to contribute to that effort, by enhancing the electronic management of patient data.

Much has been written about end-user perceptions of IT in general [1, 4, 14, 34, 39], but few studies address user evaluations of IT, specifically with regard to its application to voice activated healthcare data collection. This study focuses on the development and testing of an instrument to measure the performance and end-user evaluation of VAMTA in preventive healthcare delivery.

12.2 Background

12.2.1 *Smart Data Definition Expanded*

Our ideas about smart data [20] include three different dimensions that are expanded in the description below:

1. Performance Optimization in an Enterprise Context²

- Nontraditional systems engineering by stovepipes or verticals
- Enterprise wide scope
- Executive user leadership
- Outcome-focused

²The context in which executives address performance is truly selective in that one can choose to consider performance of a function or department, of a product or asset, or, even, of an individual. When we talk about performance optimization it is in the enterprise context versus the local context.

2. Interoperability Technology
 - Data engineering³
 - Model-driven data exchange
 - Semantic mediation⁴
 - Metadata management
 - Automated mapping tools
 - Service-Oriented Enterprise paradigm that includes Smart Data, Smart Grid, and Smart Services
 - Credentialing and privileging
3. Data-aligned Methods and Algorithms
 - Data with pointers for best practices
 - Data with built-in intelligence
 - Autonomics
 - Automated regulatory environment (ARE) and automated contracting environment (ACE)

12.2.2 Addressing the Limitations of Automated Speech Recognition

Before we present our study findings on how users adapt to Voice Activated Medical Tracking Application (VAMTA), we find it necessary to present some of the findings of speech system designers and researchers who have scrupulously analyzed some of today's most challenging problems in Automated Speech Recognition.

Speech technology has been applied, among many other vertical applications, to the medical domain, particularly emergency medical care that depends on quick and accurate access to patient background information [43]. Regardless of its vertical application, speech recognition technology is expected to play an important role in supporting real-time interactive voice communication over distributed computer data networks [29]. Yet, in spite of these demands, speech recognition engines may fall short of their promising expectations. This happens when word recognition is compromised by out of vocabulary (OOV) words, noisy texts, and other related issues affecting word error rate (WER). Neustein [36] suggests that solving some of the limitations in speech recognition accuracy rates may require a new method, called *Sequence Package Analysis* (SPA). In her work on SPA, Neustein shows

³Data engineering technologies include modeling and metadata management and smart application of known standards that account for credentialing and privileging as a dimension of security.

⁴Information modeling more fully describes data/metadata by describing the relationships between data elements as well as defining the data elements themselves. This increases the semantic content of the data, enabling the interoperability of such data by means of semantic mediation engines.

that while context-free-grammar (CFG) rules guide a speech recognizer at the lower sentence/utterance level, “SPA operates on a different plane, one especially useful in those instances when callers fail to utter the expected key word or word phrase. SPA works by examining a series of related turns and turn construction units, discretely packaged as a sequence of (conversational) interaction.” Benzeghiba et al. [7] report that “major progress is being recorded regularly on both the technology and exploitation of ASR and spoken language systems. However, there are still technological barriers to flexible solutions and user satisfaction under some circumstances. This is related to several factors, such as the sensitivity to the environment (background noise), or the weak representation of grammatical and semantic knowledge.”

Scharenborg [44] claims that the “fields of human speech recognition (HSR) and ASR both investigate parts of the speech recognition process and have word recognition as their central issue. Although these research fields appear closely related, their aims and research methods are quite different. Despite these differences there is, however, lately a growing interest in possible cross-fertilization.” Flynn and Jones [19] propose modeling combined speech enhancement for robust distributed speech recognition. Bartkova and Jouvet [3] claim that “foreign accented speech recognition systems have to deal with the acoustic realization of sounds produced by non-native speakers that does not always match with native speech models.” Rodger and Pendharkar [42] demonstrated that gender plays a role in voice recognition. Hagen et al. [23] believe that “speech technology offers great promise in the field of automated literacy and reading tutors for children. In such applications speech recognition can be used to track the reading position of the child, detect oral reading miscues, assess comprehension of the text being read by estimating if the prosodic structure of the speech is appropriate to the discourse structure of the story, or by engaging the child in interactive dialogs to assess and train comprehension. Despite such promises, speech recognition systems exhibit higher error rates for children due to variabilities in vocal tract length, formant frequency, pronunciation, and grammar.”

Cooke et al. [12] state that “robust speech recognition in everyday conditions requires the solution to a number of challenging problems, not least the ability to handle multiple sound sources. The specific case of speech recognition in the presence of a competing talker has been studied for several decades, resulting in a number of quite distinct algorithmic solutions whose focus ranges from modeling both target and competing speech to speech separation using auditory grouping principles.” Haque et al. [24] compare the performances of two perceptual properties of the peripheral auditory system, synaptic adaptation and two-tone suppression, for ASR and explore problems in an additive noise environment.

Siniscalchi et al. [45] explore a lattice rescoring approach to integrating acoustic-phonetic information into ASR and find that the rescoring process is especially effective in correcting utterances with errors in large vocabulary continuous speech recognition. Nair and Sreenivas [35] address the novel problem of jointly evaluating multiple speech patterns for ASR and training and propose solutions based on both the non-parametric dynamic time warping (DTW) algorithm, and the

parametric hidden Markov model (HMM). They show that a hybrid approach is quite effective for the application of noisy speech recognition. Torres et al. [46] present “an extension of the continuous multi-resolution entropy to different divergences and propose them as new dimensions for the pre-processing stage of a speech recognition system. This approach takes into account information about changes in the dynamics of speech signal at different scales.” Dixon et al, [15] propose harnessing graphics processors for the fast computation of acoustic likelihoods in speech recognition.

12.2.3 History of VAMTA’s Success in the Army helps its Adaptation in the Navy

Prior to 2004, few practical continuous speech recognizers were available. Most were difficult to build, or in earlier days resided on large mainframe computers, were speaker dependent, and did not operate in real time. The VAMTA which had been developed for the U.S. Army made progress in eliminating these disadvantages. VAMTA was intended to reduce the bulk, weight, and setup times of vehicle diagnostic systems while increasing their capacity and capabilities for hands-free troubleshooting. The capabilities of VAMTA were developed to allow communication with the supply and logistics structures within the Army’s common operating environment.

This effort demonstrated the use of VAMTA as a tool for a paperless method of documentation for diagnostic and prognostic results, culminating in the automation of maintenance supply actions. Voice recognition technology and existing diagnostic tools have been integrated into a wireless configuration. The result was the design of a hands-free interface between the operator and the Soldier’s On-System Repair Tool (SPORT).

The VAMTA system consisted of a microphone, a hand-held display unit, and SPORT. With this configuration, a technician could obtain vehicle diagnostic information while navigating through an Interactive Electronic Technical Manual via voice commands. By integrating paperless documentation, human expertise, and connectivity to provide user support for vehicle maintenance, VAMTA maximized U.S. Army efficiency and effectiveness.

Encouraged by the success of the Army’s VAMTA project, the U.S. Navy launched a VAMTA project of its own. The goal of the Naval Voice Interactive Device (NVID) project was to create a lightweight, portable computing device that used speech recognition to enter shipboard environmental survey data into a computer database and to generate reports automatically to fulfill surveillance requirements. Such surveillance requirements can be sine qua non in the Navy. That is, to ensure the health and safety of shipboard personnel, naval health professionals – including environmental health officers, industrial hygienists, independent duty corpsmen (IDCs), and preventive medicine technicians – must perform clinical activities and preventive medicine surveillance on a daily basis. These inspections

include, but are not limited to, water testing, heat stress, pest control, food sanitation, and habitability surveys.⁵

Typically, inspectors enter data and findings by hand onto paper forms and later transcribe these notes into a word processor or PC to create a finished report. The process of manual note-taking and entering data via keyboard into a computer database is time-consuming, inefficient, and prone to error. To remedy these problems, the Naval Shipboard Information Program was developed, allowing data to be entered into portable laptop computers while a survey is conducted [26]. However, the cramped shipboard environment, the need for mobility by inspectors, and the inability to have both hands free to type during an inspection make the use of laptop computers during a walk-around survey quite difficult. Clearly, a hands-free, space-saving mode of data entry that would also enable examiners to access and record pertinent information during an inspection was desirable. Hence, the VAMTA project was developed to fill this need.

12.2.3.1 Description of VAMTA’s Preliminary Feasibility Study aboard U.S. Navy Ships

A preliminary feasibility study, in 2004, aboard U.S. Navy ships utilized voice interactive technology to improve medical readiness. A focus group was surveyed about reporting methods in environmental and clinical inspections to develop criteria for designing a lightweight, wearable computing device with voice interactive capability. The prototype enabled quick, efficient, and accurate environmental surveillance. Existing technologies were utilized in creating the device, which was capable of storing, processing, and forwarding data to a server, as well as interfacing with other systems, including Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS), which are discussed in Section 12.2.3.2.

The voice interactive computing device included automated user prompts, enhanced data analysis, presentation, and dissemination tools in support of preventive and clinical medicine. In addition to reducing the time needed to complete inspections, the device supported local reporting requirements and enhances command-level intelligence. Limitations in the 2004 voice recognition technologies created challenges for training and user interface.

Coupling computer recognition of the human voice with a natural language processing system makes speech recognition by computers possible. By allowing data and commands to be entered into a computer without the need for typing, machine understanding of naturally spoken languages frees human hands for other tasks. Speech recognition by computers can also increase the rate of data entry, improve spelling accuracy, and permit remote access to databases utilizing wireless technology, and ease access to computer systems by those who lack proficient typing skills.

⁵ *Naval Operations Instruction 5100.19D the Navy Occupational Safety and Health Program Manual for Forces Afloat*, provides the specific guidelines for maintaining a safe and healthy work environment aboard U.S. Navy ships. Inspections performed by medical personnel ensure that these guidelines are followed.

12.2.3.2 Advantages of VAMTA aboard Ships

The 2004 VAMTA project was developed to replace existing, inefficient, repetitive medical encounter procedures with a fully automated, voice interactive system for voice-activated data input. In pursuit of this goal, the 2004 VAMTA team developed a lightweight, wearable, voice-interactive prototype capable of capturing, storing, processing, and forwarding data to a server for easy retrieval by users. The voice interactive data input and output capability of VAMTA reduced obstacles to accurate and efficient data access and reduced the time required to complete inspections. VAMTA's voice interactive technology allowed a trainee to interact with a computerized system and still have hands and eyes free to manipulate materials and negotiate his or her environment [27]. Once entered, survey and medical encounter data could be used for local reporting requirements and command-level intelligence. Improved data acquisition and transmission capabilities allowed connectivity with other systems. Existing printed and computerized surveys are voice activated and reside on the miniaturized computing device. VAMTA had been designed to allow voice prompting by the survey program, as well as voice-activated, free-text dictation. An enhanced microphone system permitted improved signal detection in noisy shipboard environments.

VAMTA technology also provided voice interactive capability documenting medical encounters using the Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS) shipboard medical database. This technology proved particularly useful in enabling medical providers to enter patient charting data rapidly and accurately. VAMTA's capability to access data from SAMS enhanced the medical providers' ability to identify trends and health hazard exposures. Researchers at the Naval Health Research Center (NHRC), San Diego, CA, are developing a clinical data analysis tool, the Epidemiological Wizard, which extracts medical data from SAMS and generates summary reports used for detecting environmental changes and early identification of disease and injury trends. In turn, these data will be analyzed to identify changes that may be indicative of an exposure to a health hazard. By integrating such data analysis tools and other emergent medical information elements with VAMTA's voice recognition technology, the VAMTA team plans to expand the ability of operational force commanders to detect disease and injury trends early, allowing quicker intervention to prevent force degradation.

Shipboard medical department personnel regularly conduct comprehensive surveys to ensure the health and safety of the ship's crew. Prior to VAMTA, surveillance data were collected and stored via manual data entry, a time-consuming process that involved typing handwritten survey findings into a word processor to produce a completed document. The VAMTA prototype was developed as a portable computer that employs voice interactive technology to automate and improve the environmental surveillance data collection and reporting process.

This 2004 prototype system was a compact, mobile computing device that included voice interactive technology, stylus screen input capability, and an indoor readable display that enables shipboard medical personnel to complete environmental

survey checklists, view reference materials related to these checklists, manage tasks, and generate reports using the collected data. The system used Microsoft Windows XP®, an operating environment that satisfies the requirement of the IT-21 Standard to which Navy ships had to conform. The major software components included initialization of VAMTA software application, application processing, database management, speech recognition, handwriting recognition, and speech-to-text capabilities. The power source for this portable unit accommodated both DC (battery) and AC (line) power options and included the ability to recharge or swap batteries to extend the system’s operational time.

The limited 2004 laboratory and field-testing described for this plan were intended to support feasibility decisions and not rigorous qualification for fielding purposes. The objectives of this plan were to describe how to:

- Validate VAMTA project objectives and system descriptions
- Assess the feasibility of voice interactive environmental tools
- Assess VAMTA prototype’s ease of use

The success of VAMTA prototype shed light on potential uses of speech recognition technology by the U.S. Navy for applications other than environmental surveillance. For example, the Navy has developed a Web-based system, the Force Health Protection System, composed of a medical database and various analytic tools for remotely or locally accessing medical data. The aggregation of the data produced a medical common operating picture. NHRC proposed to leverage this comprehensive system to develop a Web-based repository for SAMS data. These data were available to local shipboard personnel, type commanders, medical providers, and medical planners. Previously, medical personnel manually entered data documenting shipboard patient encounters into the SAMS system. To help automate this process, voice input could be incorporated into selected SAMS modules, and remote data access could be provided. Easier data input and access facilitated the investigation of higher than expected incidences of illness and/or injuries and support follow-ups measures, such as a tickler system to prompt inquiries and to check status. Voice interactive features would support the user by identifying tasks for completion and documenting the outcome of those tasks. In addition, a voice-enhanced Computer-Based Training module could be incorporated into the program to enhance training and utilization of SAMS.

To develop an appropriate voice interactive prototype system, the project team questioned end users to develop the requirement specifications. In the original 2004 study, a focus group of 14 participants (13 enlisted corpsmen, 1 medical officer) completed a survey detailing methods of completing surveys and reporting inspection results. The questionnaire addressed the needs of end users as well as their perspectives on the military utility of VAMTA. The survey consisted of 117 items ranging from nominal, yes/no answers to frequencies, descriptive statistics, rank ordering, and perceptual Likert scales. These items were analyzed utilizing the Statistical Products and Service Solutions (SPSS) statistical package. Conclusions were drawn from the statistical analysis and recommendations were suggested for development and implementation of VAMTA.

12.3 Medical Automation Case Study

We describe our specific case study below, demonstrating how users' adapt to VAMTA in the military medical environment of the U.S. Navy. Our work is premised on the TTF theory which posits that IT is more likely to have a positive impact on individual performance and be used if the capabilities of the IT match the tasks that the user must perform. Goodhue and Thompson [22] developed a measure of TTF that consists of eight factors: quality, locatability, authorization, and compatibility, ease of use/training, production timeliness, systems reliability, and relationship with users. Each factor is measured using between 2 and 10 questions with responses on a seven point scale ranging from strongly disagree to strongly agree.

The TTF asserts that for information technology to have a positive impact on individual performance, the technology: (1) must be utilized and (2) must be a good fit with the tasks it supports.

The VAMTA case study was initiated in order to study this voice application with medical end-users. This first phase of the case study, which we referred to as the pilot, provided us with face validity. The case used in the in-depth study demonstrated that the perceptions of end-users can be measured and an evaluation of the system from a conceptual viewpoint can be documented – in order to determine the scope of this non-traditional application.

The case survey results were analyzed using the Statistical Package for the Social Sciences (SPSS) data analysis tool to determine whether TTF, along with individual characteristics, will have an impact on user evaluations of VAMTA. The case modified the original TTF model for adequate domain coverage of medical patient-care applications, and provides the underpinnings for a subsequent, higher level study of nationwide medical personnel. Follow-on studies will be conducted to investigate performance and user perceptions of the VAMTA system under actual medical field conditions.

Here are the fundamental aspects of our empirical study of user adaptation to VAMTA in the Navy:

- *Background:* the customer is the Joint Military Medical Command of the US Department of Defense.
- *Goals:* validate that Voice Activated Medical Technology Application produces reliable and accurate results while affording cost and time savings and convenience advantages.
- *Decision:* should the Voice Activated Medical Tracking be applied throughout the military medical community?
- *IT Support:* Provide a methodology and algorithms for validation. Help standardize data capture, recording, and processing.

12.3.1 Electronic Information Sharing and Connectivity

As the technological infrastructure of organizations becomes ever more complex (Henderson, [25]), IT is increasingly being used to improve coordination of activities

within and across organizations (Cash and Konsynski, [11]). Computers and Video networks provide long-distance healthcare through medical connectivity, allowing doctors to interact with each other and ancillary medical personnel through e-mail, Video, and audio means. A difficult patient case in a rural area, or on shipboard, can be given expert specialist attention simply by using “distance” medicine. Not only can patient records, text, and documents be transmitted instantaneously via electronic means, but live Video, x-rays, and other diagnostic parameters can be discussed in an interactive manner with live discussions.

As the availability of external consultative services increases, information sharing and connectivity are becoming increasingly important. Connectivity allows diagnoses to be made in remote locations using electronic means. What is more, information sharing decreases the chances that mistakes will be made in a healthcare setting. In the last analysis, connectivity leads to shared-care, characterized by continued, coordinated, and integrated activities of a multitude of people from various institutions applying a variety of methods in different time frames – all of which adds up to a combined effort to aid patients medically, psychologically, and socially in the most beneficial ways [17].

In addition to electronic information sharing and connectivity, IT has been and continues to be widely used for staff and equipment scheduling in healthcare settings. IT-based scheduling can lower healthcare costs and improve the utilization of physical and human resources. Scheduling using statistical, time series and regression analysis is conducted to achieve lower costs through rationing assets (e.g., ambulatory service and real-time forecasting of resources) [37].

12.3.2 Setting the Stage

The purpose of this 2009 study was to develop and test a valid survey instrument for measuring user evaluations of VAMTA in preventive healthcare. The findings of a 2004 pilot study testing the instrument were used in a preliminary assessment of the effectiveness of VAMTA system and the applicability of TTF to VAMTA.

The development of the instrument was carried out in two stages. The first stage was item creation. The objective of this first stage was to ensure face and content validity of the instrument. An item pool was generated by interviewing two end-users of IT, obtained from a pool of medical technicians. The end-users were given training on the module for two days and invited to participate in the study. These subjects were selected for reasons of geographical proximity of the sample and, in many cases, the existence of personal contacts onboard ship.

An interview was also conducted with one of the authors of this study, who has approximately 10 years of experience as an IT end-user. In addition, the domain coverage of the developed pool of items was assessed by three other end-users from three different ship environments covered in the survey. None of the end-users, who were a part of the scale development, completed the final survey instrument. All the items were measured on a five-point Likert scale ranging from “strongly agree” to “strongly disagree.” Next, the survey instrument was utilized in a study in which end-users tested VAMTA. While

the pilot study provided face validity, this study demonstrates that the perceptions of end-users can be measured, and the system evaluated from a conceptual viewpoint. A total of 33 end-users were used in this phase to test VAMTA. They reported their perceptions of VAMTA in the survey instrument, which was provided after their training and testing. The pilot study results were analyzed using SPSS and Microsoft Excel to determine whether TTF, along with individual characteristics, had an impact on user evaluations of VAMTA. For the study, the original TTF model was modified to ensure adequate domain coverage of medical and preventive healthcare applications.

12.3.2.1 Instrument Development and Measurement of Variables

The IT construct used in the pilot study focused on the use of VAMTA to support preventive medicine applications. Construction of the survey instrument was based in part on Akaike's (2) information criterion (AIC) and Bozdogan's (5) consistent information criterion (CAIC).

12.3.3 Testing Procedure

In the original 2004 study, each test subject was shown a demonstration of VAMTA application prior to testing. Test subjects were then required to build a new user account and speech profile. Subjects ran through the application once using a test script to become familiar with the application. Next, the test subjects went through the application again while being videotaped. No corrections were made to dictated text during the Videotaping. This allowed the tracking of voice dictation accuracy for each user with a new speech profile. Test subjects completed the entire process in an average of two hours.

Afterward, each test subject completed a questionnaire to determine user demographics, utility and quality performance, system interface, hardware, information management, and overall system satisfaction and importance. This survey instrument also allowed the test subjects to record any problems and suggest improvements to the system. Problems recorded by test subjects in the survey instrument were also documented in the Bug Tracking System by the test architect for action by the VAMTA development team.

12.3.3.1 Pilot Study Test Results

Prior to the 2009 study, a pilot test was performed in 2004, in order to test the feasibility of VAMTA for medical encounters. In the 2004 study, the performance of VAMTA during testing was measured in terms of voice accuracy, voice accuracy total errors, duration with data entry by voice, and duration with data entry by keyboard and mouse. Viewed together, these statistics provide a snapshot of the accuracy, speed, and overall effectiveness of the VAMTA system.

Each test subject’s printout was compared with a test script printout for accuracy. When discrepancies occurred between the subject’s printout and the test script, the printouts were compared with the video recordings to determine whether the test subjects said the words properly, stuttered or mumbled words, and/or followed the test script properly. Misrecognitions occurred when the test subject said a word properly but the speech program recorded the wrong word.

The accuracy of voice recognition, confirmed by videotaped records of test sessions, averaged 97.6%, with six misrecognitions (Table 12.1). The minimum average voice recognition was 85%, with 37 misrecognitions. The maximum average voice recognition was 99.6%, with one misrecognition. Median voice recognition was 98.4%, with four misrecognitions.

Total errors include both misrecognitions and human errors. Human errors occurred when a test subject mispronounced a word, stuttered, or mumbled. The total accuracy rate of VAMTA was 95.4% (Table 12.2). Human error accounted for 2.2% of the total errors within the application.

The duration of each test subject’s voice dictation was recorded to determine the average length of time required to complete a medical encounter (the doctor’s entry of patient information into the system) while using VAMTA. The average time required to complete a VAMTA medical encounter in which data entry was conducted by voice was 8 min and 31 s (Table 12.3). The shortest time was 4 min and 45 s, and the longest time was 23 min and 51 s.

Table 12.1 VAMTA voice accuracy during testing

| | Misrecognitions with video | # Correct with punctuation and video | % Accurate with video |
|---------|----------------------------|--------------------------------------|-----------------------|
| Average | 6 | 241 | 97.6 |
| Minimum | 1 | 210 | 85.0 |
| Maximum | 37 | 246 | 99.6 |
| Median | 4 | 243 | 98.4 |
| Males | 22 | | |
| Females | 11 | | |
| Count | 33 | | |

Table 12.2 VAMTA voice accuracy total errors during testing

| Total errors | |
|------------------|-------|
| Average Accurate | 95.4% |
| Minimum Accurate | 85.0% |
| Maximum Accurate | 99.2% |
| Median | 96.0% |
| Count | 33 |

Table 12.3 Medical encounter duration with data entry by voice

| | |
|----------------------|---------|
| Average Time – voice | 0:08:31 |
| Minimum Time – voice | 0:04:54 |
| Maximum Time – voice | 0:32:51 |
| Median | 0:06:59 |

While the majority of test subjects entered medical encounter information into VAMTA only by voice, several test subjects entered the same medical encounter information using a keyboard and mouse. The average time required to complete a medical encounter in which data entry was conducted with keyboard and mouse was 15 min and 54 s (Table 12.4). The shortest time was 7 min and 52 s, and the longest time was 24 min and 42 s.

The average duration of sessions in which data entry was performed by voice dictation was compared to the average duration of sessions in which data entry was performed with a keyboard and mouse. On average, less time was required to complete the documentation of a medical encounter using VAMTA when data entry was performed by voice instead of with a keyboard and mouse.

The average time saved using voice versus a keyboard and mouse was 7 min and 52 s per medical encounter. The duration of each medical encounter included the dictation and printing of the entire Chronological Record of Medical Care form, a Poly Prescription form, and a Radiologic Consultation Request/Report form (Tables 12.5–12.13).

Table 12.4 Medical encounter duration with data entry by keyboard and mouse

| | |
|----------------------------|---------|
| Average time with keyboard | 0:15:54 |
| Minimum time with keyboard | 0:07:52 |
| Maximum time with keyboard | 0:24:42 |
| Median | 0:15:31 |

Table 12.5 VAMTA T&E data human errors

| Subject number | Total with punc | Human errors | # Right minus human errors | % Accurate human errors | Sex |
|----------------|-----------------|--------------|----------------------------|-------------------------|-----|
| 1 | 247 | 0 | 247 | 100.0 | F |
| 2 | 247 | 12 | 235 | 95.1 | F |
| 3 | 247 | 12 | 235 | 95.1 | F |
| 4 | 247 | 3 | 244 | 98.8 | M |
| 5 | 247 | 0 | 247 | 100.0 | F |
| 6 | 247 | 0 | 247 | 100.0 | F |
| 7 | 247 | 2 | 245 | 99.2 | M |
| 8 | 247 | 2 | 245 | 99.2 | M |
| 9 | 247 | 9 | 238 | 96.4 | M |
| 10 | 247 | 16 | 231 | 93.5 | M |
| 11 | 247 | 15 | 232 | 93.9 | M |
| 12 | 247 | 1 | 246 | 99.6 | F |
| 13 | 247 | 0 | 247 | 100.0 | F |
| 14 | 247 | 10 | 237 | 96.0 | F |
| 15 | 247 | 13 | 234 | 94.7 | M |
| 16 | 247 | 7 | 240 | 97.2 | F |
| 17 | 247 | 6 | 241 | 97.6 | M |
| 18 | 247 | 14 | 233 | 94.3 | M |
| 19 | 247 | 1 | 246 | 99.6 | M |
| 20 | 247 | 3 | 244 | 98.8 | F |

(continued)

Table 12.5 (continued)

| Subject number | Total with punc | Human errors | # Right minus human errors | % Accurate human errors | Sex |
|----------------|-----------------|--------------|----------------------------|-------------------------|-----|
| 21 | 247 | 9 | 238 | 96.4 | M |
| 22 | 247 | 7 | 240 | 97.2 | M |
| 23 | 247 | 6 | 241 | 97.6 | M |
| 24 | 247 | 3 | 244 | 98.8 | M |
| 25 | 247 | 0 | 247 | 100.0 | M |
| 26 | 247 | 4 | 243 | 98.4 | F |
| 27 | 247 | 1 | 246 | 99.6 | M |
| 28 | 247 | 2 | 245 | 99.2 | M |
| 29 | 247 | 11 | 236 | 95.5 | M |
| 30 | 247 | 3 | 244 | 98.8 | M |
| 31 | 247 | 1 | 246 | 99.6 | M |
| 33 | 247 | 1 | 246 | 99.6 | M |

Table 12.6 VAMTA T&E aggregate data human errors

| Average accurate | Minimum accurate | Maximum accurate | Median | Count |
|------------------|------------------|------------------|--------|-------|
| 97.8% | 93.5% | 100.0% | 98.8% | 33 |

Table 12.7 VAMTA T&E data total errors

| Subject number | Total | Total errors | # Right | % Accurate | Sex |
|----------------|-------|--------------|---------|------------|-----|
| 1 | 247 | 13 | 234 | 94.7 | F |
| 2 | 247 | 13 | 234 | 94.7 | F |
| 3 | 247 | 14 | 233 | 94.3 | F |
| 4 | 247 | 6 | 241 | 97.6 | M |
| 5 | 247 | 4 | 243 | 98.4 | F |
| 6 | 247 | 4 | 243 | 98.4 | F |
| 7 | 247 | 8 | 239 | 96.8 | M |
| 8 | 247 | 4 | 243 | 98.4 | M |
| 9 | 247 | 13 | 234 | 94.7 | M |
| 10 | 247 | 25 | 222 | 89.9 | M |
| 11 | 247 | 22 | 225 | 91.1 | M |
| 12 | 247 | 2 | 245 | 99.2 | F |
| 13 | 247 | 37 | 210 | 85.0 | F |
| 14 | 247 | 26 | 221 | 89.5 | F |
| 15 | 247 | 17 | 230 | 93.1 | M |
| 16 | 247 | 12 | 235 | 95.1 | F |
| 17 | 247 | 8 | 239 | 96.8 | M |
| 18 | 247 | 19 | 228 | 92.3 | M |
| 19 | 247 | 6 | 241 | 97.6 | M |
| 20 | 247 | 9 | 238 | 96.4 | F |
| 21 | 247 | 20 | 227 | 91.9 | M |

(continued)

Table 12.7 (continued)

| Subject number | Total | Total errors | # Right | % Accurate | Sex |
|----------------|-------|--------------|---------|------------|-----|
| 22 | 247 | 12 | 235 | 95.1 | M |
| 23 | 247 | 11 | 236 | 95.5 | M |
| 24 | 247 | 10 | 237 | 96.0 | M |
| 25 | 247 | 3 | 244 | 98.8 | M |
| 26 | 247 | 8 | 239 | 96.8 | F |
| 27 | 247 | 8 | 239 | 96.8 | M |
| 28 | 247 | 4 | 243 | 98.4 | M |
| 29 | 247 | 14 | 233 | 94.3 | M |
| 30 | 247 | 10 | 237 | 96.0 | M |
| 31 | 247 | 5 | 242 | 98.0 | M |
| 33 | 247 | 3 | 244 | 98.8 | M |

Table 12.8 VAMTA T&E aggregate data total errors

| Average accurate | Minimum accurate | Maximum accurate | Median | Count |
|------------------|------------------|------------------|--------|-------|
| 95.4% | 85.0% | 99.2% | 96.0% | 33 |

Table 12.9 VAMTA T&E data female

| Subject number | Total punc | Miss recognitions with video | # Right with punc & video | % Accurate with video | Time started Sex | Time voice stopped | Time voice to complete |
|----------------|------------|------------------------------|---------------------------|-----------------------|------------------|--------------------|------------------------|
| 1 | 247 | 13 | 234 | 94.7 | F 13:35:43 | 14:08:34 | 0:32:51 |
| 2 | 247 | 1 | 246 | 99.6 | F 14:26:53 | 14:32:18 | 0:05:25 |
| 3 | 247 | 2 | 245 | 99.2 | F 10:49:42 | 11:01:15 | 0:11:33 |
| 5 | 247 | 4 | 243 | 98.4 | F 11:18:13 | 11:23:27 | 0:05:14 |
| 6 | 247 | 4 | 243 | 98.4 | F 13:35:02 | 13:42:16 | 0:07:14 |
| 12 | 247 | 1 | 246 | 99.6 | F 14:28:28 | 14:33:37 | 0:05:09 |
| 13 | 247 | 37 | 210 | 85.0 | F 10:49:39 | 11:01:28 | 0:11:49 |
| 14 | 247 | 16 | 231 | 93.5 | F 9:45:02 | 10:07:50 | 0:22:48 |
| 16 | 247 | 5 | 242 | 98.0 | F 10:41:12 | 10:47:18 | 0:06:06 |
| 20 | 247 | 6 | 241 | 97.6 | F 10:45:40 | 10:51:11 | 0:05:31 |
| 26 | 247 | 4 | 243 | 98.4 | F 11:06:50 | 11:12:58 | 0:06:08 |

Table 12.10 VAMTA T&E aggregate data female

| Average accurate | Minimum accurate | Maximum accurate | Median | Total females |
|------------------|------------------|------------------|--------|---------------|
| 96.6% | 85.0% | 99.6% | 98.4% | 11 |

Table 12.11 VAMTA T&E Aggregate data for max and min voice times

| Average time voice | Minimum time voice | Maximum time voice | Median |
|--------------------|--------------------|--------------------|---------|
| 0:10:53 | 0:05:09 | 0:32:51 | 0:06:08 |

Table 12.12 VAMTA T&E data total errors

| Subject number | Total | Total errors | # Right | % Accurate | Sex |
|----------------|-------|--------------|---------|------------|-----|
| 1 | 247 | 13 | 234 | 94.7 | F |
| 2 | 247 | 13 | 234 | 94.7 | F |
| 3 | 247 | 14 | 233 | 94.3 | F |
| 4 | 247 | 6 | 241 | 97.6 | M |
| 5 | 247 | 4 | 243 | 98.4 | F |
| 6 | 247 | 4 | 243 | 98.4 | F |
| 7 | 247 | 8 | 239 | 96.8 | M |
| 8 | 247 | 4 | 243 | 98.4 | M |
| 9 | 247 | 13 | 234 | 94.7 | M |
| 10 | 247 | 25 | 222 | 89.9 | M |
| 11 | 247 | 22 | 225 | 91.1 | M |
| 12 | 247 | 2 | 245 | 99.2 | F |
| 13 | 247 | 37 | 210 | 85.0 | F |
| 14 | 247 | 26 | 221 | 89.5 | F |
| 15 | 247 | 17 | 230 | 93.1 | M |
| 16 | 247 | 12 | 235 | 95.1 | F |
| 17 | 247 | 8 | 239 | 96.8 | M |
| 18 | 247 | 19 | 228 | 92.3 | M |
| 19 | 247 | 6 | 241 | 97.6 | M |
| 20 | 247 | 9 | 238 | 96.4 | F |
| 21 | 247 | 20 | 227 | 91.9 | M |
| 22 | 247 | 12 | 235 | 95.1 | M |
| 23 | 247 | 11 | 236 | 95.5 | M |
| 24 | 247 | 10 | 237 | 96.0 | M |
| 25 | 247 | 3 | 244 | 98.8 | M |
| 26 | 247 | 8 | 239 | 96.8 | F |
| 27 | 247 | 8 | 239 | 96.8 | M |
| 28 | 247 | 4 | 243 | 98.4 | M |
| 29 | 247 | 14 | 233 | 94.3 | M |
| 30 | 247 | 10 | 237 | 96.0 | M |
| 31 | 247 | 5 | 242 | 98.0 | M |
| 33 | 247 | 3 | 244 | 98.8 | M |
| 34 | 247 | 8 | 239 | 96.8 | M |

Table 12.13 VAMTA T&E Aggregate Data Male

| Average accurate | Minimum accurate | Maximum accurate | Median | Total Males |
|------------------|------------------|------------------|--------|-------------|
| 93.4% | 89.9% | 96.8% | 95.5% | 22 |

12.3.3.2 Related Work and Theoretical Framework for our In-Depth Study

While descriptive statistics formed the basis for the original 2004 pilot study to establish face validity, we based our ensuing in-depth 2009 study on the TTF model (Fig. 12.1). This is a popular model for assessing user evaluations of information systems. The central premise for the TTF model is that “users will give evaluations

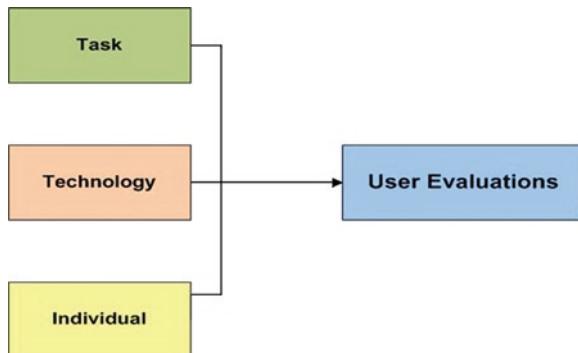


Fig. 12.1 Task technology fit model

based on the extent to which systems meet their needs and abilities” [22]. For the purpose of our study, we defined user evaluations as the user perceptions of the fit of systems and services they use, based on their personal task needs [22].

The TTF model represented in Fig. 12.1 is very general, thus using it for a particular setting requires special consideration. Among the three factors appearing in Fig. 12.1 (Task, Technology and Individual) that determine user evaluations of information systems, *technology* is the most complex factor to measure in healthcare. Technology in healthcare is used primarily for reporting, electronic information sharing and connectivity, and staff and equipment scheduling.

Reporting is important in a healthcare setting because patient lives depend on accurate and timely information. Functional departments within the healthcare facility must be able to access and report new information in order to respond properly to changes in the healthcare environment [31].

Four types of information are reported in a healthcare facility:

- Scientific and technical information
- Patient-care information
- Customer satisfaction information
- Administrative information [40,41].

Scientific and technical information provides the knowledge base for identifying, organizing, retrieving, analyzing, delivering, and reporting clinical and managerial journal literature, reference information, and research data for use in designing, managing, and improving patient-specific and departmental processes [28].

Patient-care information is specific data and information on patients that is essential for maintaining accurate medical records of the patients’ medical histories and physical examinations. Patient-specific data and information are critical to tracking all diagnostic and therapeutic procedures and tests. Maintaining accurate information about patient-care results and discharges is imperative to delivering quality healthcare [5].

Customer satisfaction information is gathered from external customers, such as a patient and his or her family and friends. Customer satisfaction information is gathered from surveys and takes into account socio-demographic characteristics,

physical and psychological status, attitudes, and expectations concerning medical care, the outcome of treatment, and the healthcare setting [32].

The administrative information that is reported in a healthcare facility is essential for formulating and implementing effective policies both at the organizational and departmental level. Administrative information is necessary to determine the degree of risk involved in financing expansion of services [16].

12.3.3.3 Survey Response Analysis

Whereas in the 2004 pilot study the effectiveness of VAMTA during its test phase was estimated purely from statistics capturing *voice accuracy* and *duration of medical encounters*, in the 2009 in-depth study the applicability of the TTF model to the VAMTA system was determined by analyzing *end-user survey instrument responses*. Multiple regression analysis revealed the effects of VAMTA utility, quality performance, task characteristics, and individual characteristics on user evaluations of VAMTA (Table 12.14).

The results indicate that overall end-user evaluations of VAMTA are consistent with the TTF model. The *F* value was 6.735 and the model was significant at the $p=0.001$ level of significance. The *R*-square for the model was 0.410. This indicates that model-independent variables explain 41% of the variance in the dependent variable. The individual contributions of each independent variable factor are shown in Table 12.15.

While the Table 12.14 data reveals the suitability of the TTF model, Table 12.15 reveals another finding. Based on the data shown in Table 12.15, according to the TTF study, user evaluations of VAMTA, utility factors such as navigation, application and operation as well as quality performance factors such as ease of use and understandable, are the *major* factors that affect the management of information by VAMTA.

Table 12.14 One-way ANOVA table for regression analysis

| Degrees of freedom | Source | Sum of SQ. | Mean SQ. | F value | <i>p>F*</i> |
|--------------------|------------|------------|----------|---------|----------------|
| 4 | Regression | 2.664 | 0.666 | 6.735 | .001 |
| 29 | Residual | 2.867 | 0.099 | | |
| 33 | Total | 5.531 | | | |

*Significant at $p=0.001$

Table 12.15 Individual contribution of the study variables to the dependent variable

| Source | Degrees of freedom | Sum of Sq. | Mean SQ. | <i>F</i> value | <i>p>F</i> |
|----------------|--------------------|------------|----------|----------------|---------------|
| | | | | | 0.022** |
| Ease of Use | 1 | 1.577 | 1.577 | 5.834 | 0.007** |
| Navigation | 1 | 2.122 | 2.122 | 8.459 | 0.0000* |
| Application | 1 | 0.211 | 3.80 | 4.59 | 0.0000* |
| Operation | 1 | 0.912 | 0.912 | 3.102 | 0.008** |
| Understandable | 1 | 0.965 | 0.965 | 3.196 | 0.0085** |

*Significant at $p = 0.001$, ** Significant at $p = 0.05$

12.3.3.4 Survey Results

The following discussion presents the results of the survey and how the information was incorporated into the prototype.

The commands and ranks of these participants are shown in Table 12.16. These participants possessed varying clinical experience while assigned to deployed units (ships and Fleet Marine Force), including IDCs, preventive medicine, lab technicians, and aviation medicine.

Environmental Health and Preventive Medicine Afloat

In the first section of the questionnaire, inspectors were asked about the methods they used to record findings while conducting an inspection (see Table 12.17).

Response to this section of the questionnaire was limited. The percentage of missing data ranged from 7.1% for items such as habitability and food sanitation safety to 71.4% for mercury control and 85.7% for polychlorinated biphenyls. The majority of inspectors relied on preprinted checklists. Fewer inspections were conducted utilizing handwritten reports. Only 7.1% of the users recorded their findings on a laptop computer for inspections focusing on radiation protection, workplace monitoring, food sanitation safety, and habitability.

In addition to detailing their methods of recording inspection findings, the focus group participants were asked to describe the extensiveness of their notes during surveys. The results ranged from “one to three words in a short phrase” (35.7%) to “several short phrases, up to a paragraph” (64.3%). No respondents claimed to have used “extensive notes of more than one paragraph.” The participants were also asked how beneficial voice dictation would be while conducting an inspection.

Table 12.16 VAMTA focus group participants

| Command | Rank/Rate |
|---|------------------|
| Navy Environmental Preventative Medicine Unit-5 | HM2 ¹ |
| Navy Environmental Preventative Medicine Unit-5 | HM1 ² |
| Navy Environmental Preventative Medicine Unit-5 | HM3 ³ |
| Commander Submarine Development Squadron Five | HMCS |
| Naval School of Health Sciences, San Diego | HMCS |
| Naval School of Health Sciences, San Diego | HM1 |
| Naval School of Health Sciences, San Diego | HMC |
| Commander, Amphibious Group-3 | HMC |
| Commander, Amphibious Group-3 | HMC |
| USS CONSTELLATION (CV-64) | HMC |
| USS CONSTELLATION (CV-64) | HMC |
| Commander, Naval Surface Force Pacific | HMCS |
| Commander, Naval Surface Force Pacific | HMCS |
| Regional Support Office, San Diego | CDR |

¹HM2, Hospitalman Second Class HMC. Hospitalman Chief

²HM1, Hospitalman First Class HMCS, Hospitalman Senior Chief

³HM3, Hospitalman Third Class CDR, Commander

Table 12.17 Methods of recording inspection findings

| Inspections | Handwritten (%) | Preprinted check lists (%) | Laptop Computer | Missing (%) |
|--|-----------------|----------------------------|-----------------|-------------|
| Asbestos | 14.3 | 50.0 | 0 | 35.7 |
| Heat stress | 14.3 | 71.4 | 0 | 14.3 |
| Hazardous materials | 21.4 | 50.0 | 0 | 28.6 |
| Hearing conservation | 21.4 | 64.3 | 0 | 14.3 |
| Sight conservation | 7.1 | 71.4 | 0 | 21.4 |
| Respiratory conservation | 0 | 71.4 | 0 | 28.6 |
| Electrical safety | 14.3 | 50.0 | 0 | 35.7 |
| Gas-free engineering | 14.3 | 28.6 | 0 | 57.1 |
| Radiation protection | 7.1 | 28.6 | 7.1 | 57.1 |
| Lead control | 0 | 64.3 | 0 | 35.7 |
| Tag-out program | 7.1 | 50.0 | 0 | 42.9 |
| Personal protective equipment | 7.1 | 42.9 | 0 | 50.0 |
| Mercury control | 0 | 28.6 | 0 | 71.4 |
| PCBs | 0 | 14.3 | 0 | 85.7 |
| Man-made vitreous fibers | 7.1 | 28.6 | 0 | 64.3 |
| Blood-borne pathogens | 0 | 50.0 | 0 | 50.0 |
| Workplace monitoring | 0 | 42.9 | 7.1 | 50.0 |
| Food sanitation safety | 14.3 | 71.4 | 7.1 | 7.1 |
| Habitability | 28.6 | 57.1 | 7.1 | 7.1 |
| Potable water, halogen/bacterial testing | 35.7 | 57.1 | 0 | 7.1 |
| Wastewater systems | 21.4 | 50.0 | 0 | 28.6 |
| Other | 0 | 0 | 0 | 100 |

PCBs, polychlorinated biphenyls, pentachlorobenzole

Those responding that it would be “very beneficial” (71.4%) far outweighed those responding that it would be “somewhat beneficial” (28.6%). No respondents said that voice dictation would be “not beneficial” in conducting an inspection. In another survey question, participants were asked if portions of their inspections were done in direct sunlight. The “yes” responses of (92.9%) of those with a computer who worked in direct sunlight were far more prevalent than the “no” responses (7.1%) with a computer who did not work in direct sunlight.

Participants also described the types of reference material needed during inspections. The results are shown in Table 12.18.

“Yes” responses ranged from a low of 28.6% for procedure description information to 78.6% for current checklist in progress information. When asked how often they utilized reference materials during inspections, no participants chose the response “never.” Other responses included “occasionally” (71.4%), “frequently” (21.4%) and “always” (7.1%). In another survey question, participants were asked to describe their methods of reporting inspection results, which included the following: preparing the report using SAMS (14.8%), preparing the report using word processing other than SAMS (57.1%), and preparing the report using both SAMS and word

Table 12.18 Types of reference information needed during inspections

| Information | Yes (%) | No (%) |
|--|---------|--------|
| Current checklist in progress | 78.6 | 21.4 |
| Bureau of medicine instructions | 71.4 | 28.6 |
| Naval operations instructions | 71.4 | 28.6 |
| Previously completed reports for historical references | 71.4 | 28.6 |
| Exposure limit tables | 57.1 | 42.9 |
| Technical publications | 57.1 | 42.9 |
| Type commander instructions | 50.0 | 50.0 |
| Local INSTRUCTIONS | 42.9 | 57.1 |
| Procedures descriptions | 28.6 | 71.4 |
| Other | 21.4 | 78.6 |

processing (28.6%). No respondents reported using handwritten or other methods of reporting inspection results. Participants were also asked how they distributed final reports. The following results were tabulated: hand-carry (21.4%); guard mail (0%); download to disk and mail (7.1%); Internet e-mail (64.3%); upload to server (0%); file transfer protocol (FTP) (0%); and other, not specified (7.1%). When asked if most of the problems or discrepancies encountered during an inspection could be summarized using a standard list of “most frequently occurring” discrepancies, 100% of respondents answered “yes.” The average level of physical exertion during inspections was reported as Light by 42.9% of respondents, Moderate by 50.0% of respondents and Heavy by 7.1% of respondents. Survey participants were also asked to describe their level of proficiency at ICD-9 CM (Department of Health and Human Services, 1989). An expert level of proficiency was reported 7.1% of the time. Other responses included “competent” (14.3%), “good” (28.6%), “fair” (28.6%), and “poor” (7.1%). Missing data made up 14.3% of the responses.

Shipboard Computer Software and Hardware

In the second section of the questionnaire, end users addressed characteristics of shipboard medical departments, VAMTA, medical encounters, and SAMS. When asked if their medical departments were connected to a local area network (LAN), respondents answered as follows: “yes” (71.4%), “no” (7.1%), and “uncertain” (14.3%). Missing responses totaled 7.1%. Participants asked if their medical departments had LANs of their own responded “yes” (14.3%), “no” (57.1%), and “uncertain” (21.4%). Another 7.1% of responses to this question were missing. When asked if their medical departments had access to the Internet, participants responded “yes, in medical department” (85.7%); and “yes, in another department” (7.1%). Another 7.1% of responses were missing.

Various methods for transmitting medical data from ship to shore were also examined in the survey. It was found that 78.6% of those surveyed said they had used Internet e-mail, while 14.3% said that they had downloaded data to a disk and mailed it. No users claimed to have downloaded data to a server or utilized FTP for this purpose. Missing responses totaled 7.1%.

Table 12.19 shows respondents’ rankings of the desirable features of the device. The Likert scale had a range of 1, “very desirable” to 7, “not very desirable”.

“Voice activation dictation” and “durability” were tied for the top ranking indication that few changes were necessary in these areas. “Wearable in front or back” and “earphones” were tied for lowest ranking indicating that end users were not very concerned with these details. “Voice prompting for menu navigation” and “LAN connectivity” were the number 3 and 4 choices, respectively, indicating that perhaps some more thought needed to be put into these areas. Respondents’ rankings of medical encounter types are shown in Table 12.20.

According to the rankings, routine sick call is the type of medical encounter for which voice automation is most desirable, followed by physical exams and emergency care. Immunizations and medical evacuations ranked lowest on the list, probably because the end users had alternate methods for handling these tasks.

Participants ranked ancillary services for voice automation desirability (Table 12.21).

Pharmacy and laboratory services were the most desired because these tasks lent themselves better to VAMTA technology. Of the respondents, 92.9% also indicated that voice automation would enhance the cataloging and maintenance of the Authorized Medical Allowance List, while only 7.14% answered “no.”

Table 12.19 Ranking of device features

| Feature | Average | Rank |
|-------------------------------------|---------|---------|
| Voice activated dictation | 2.64 | 1(tie) |
| Durability | 2.64 | 1 (tie) |
| Voice prompting for menu navigation | 2.93 | 3 |
| LAN connectivity | 4.21 | 4 |
| Belt or harness wearability | 4.57 | 5 |
| Wireless microphone | 5.29 | 6 |
| Touch pad/screen | 5.93 | 7 |
| Earphones | 6.14 | 8 (tie) |
| Wearable in front or back | 6.14 | 8 (tie) |

LAN, local area network

Table 12.20 Ranking medical encounter types

| Medical encounter types | Average | Rank |
|-------------------------|---------|--------|
| Routine sick call | 1.43 | 1 |
| Physical EXAMS | 2.79 | 2(tie) |
| Emergency care | 2.79 | 2(tie) |
| Consultation | 3.07 | 4 |
| Immunizations | 3.43 | 5 |
| Medical evacuations | 4.57 | 6 |

Table 12.21 Ranking ancillary services

| Ancillary Services | Average | Rank |
|--------------------|---------|------|
| Pharmacy | 1.29 | 1 |
| Laboratory | 1.36 | 2 |
| Physical therapy | 3.14 | 4 |
| Radiological | 2.79 | 3 |

Table 12.22 Ranking SAMS modules

| SAMS Modules | Average | Rank |
|-----------------------------------|---------|------|
| Master tickler | 1.71 | 1 |
| Medical encounters | 1.93 | 2 |
| Supply management | 2.14 | 3 |
| Occupational/environmental health | 2.64 | 4 |
| Training management | 3.50 | 5 |
| Radiation health | 3.86 | 6 |
| Periodic duties | 4.43 | 7 |
| Smart card review | 5.50 | 8 |

Table 12.23 Elements used in reports to identify inspected areas

| Identifying element | Yes | No |
|---------------------|------|------|
| Compartment number | 57.1 | 42.9 |
| Department | 57.1 | 42.9 |
| Name of area | 85.7 | 14.3 |
| Other | 0 | 100 |

Participants in the survey were also asked to rank SAMS modules according to frequency of use (Table 12.22).

“Master Tickler,” “Medical Encounters” and “Supply Management” were the most frequently used modules because they lend themselves to VAMTA task technology fit. In another question, participants rated their computer efficiency. Just 14.3% rated their computer efficiency as “expert,” while 42.9% chose “competent.” “Good” and “fair” were each selected by 21.4% of respondents.

Participants reportedly used “name of area” as the most used element (85.7%) to identify an inspected area (Table 12.23).

Table 12.24 provides respondents’ rankings of the areas of environmental surveillance in which voice automation would be of the greatest value. According to this rank ordering, “Food Sanitation Safety” would most benefit from voice automation. “Heat Stress” and “Potable Water, Halogen” were also popular choices.

Professional Opinions

In the third section of the survey, participants were asked which attributes of VAMTA they would find most desirable (Table 12.25). A regression analysis was performed on the Likert scales.

It was hypothesized that in an automated system, the reduction of data entry, the availability of an online tutorial, the availability of a lightweight device for documenting encounters, the reduction of paperwork, and the ability to see an overview instantly would favorably reduce the difficulty in assigning ICD-9 codes (medical and psychiatric diagnoses for patient billing) The regression yielded a *p* value of 0.0220. This is below the 0.05 threshold and gives us a 95% confidence interval that there may be a correlation between the dependent and independent variables. The

Table 12.24 Surveillance areas benefiting from voice automation

| Areas | Average | Rank |
|---|---------|------|
| Food sanitation safety | 1.21 | 1 |
| Heat Stress | 3.29 | 2 |
| Potable water, halogen | 3.86 | 3 |
| Habitability | 4.14 | 4 |
| Potable water, bacterial | 4.21 | 5 |
| Inventory TOOL | 4.43 | 6 |
| Hazard-specific programs with checklist | 4.86 | 7 |

Table 12.25 Frequencies of Desirable Attributes

| Opinion | Strongly agree (%) | Agree (%) | Unsure (%) | Disagree (%) | Strongly disagree (%) |
|-----------------------------------|--------------------|-----------|------------|--------------|-----------------------|
| Care for patients | 71.4 | 28.6 | | | |
| Reduce data entries | 21.4 | 71.4 | 7.1 | | |
| Reduce paperwork | 14.3 | 57.1 | 14.3 | 14.3 | |
| Conduct outbreak analysis | 21.4 | 35.7 | 21.4 | 21.4 | |
| On-line tutorial | 14.3 | 57.1 | 21.4 | 7.1 | |
| Lightweight device | 21.4 | 71.4 | 7.1 | | |
| See an overview | 28.6 | 50.0 | 14.3 | 7.1 | |
| Automated ICD-9-CM | 35.7 | 42.9 | 7.1 | 14.3 | |
| Difficulties using ICD-9-CM codes | 14.2 | 28.6 | 28.6 | 28.6 | |
| ICD-9-CM, | | | | | |

R square value of 0.759 tells us that about three fourths of the dependent variable is explained by the independent variables. This indicates that while there is a good fit between the dependent and independent variables, there is no multi-collinearity. Correlation coefficients reported no multi-collinearity among the independent variables. Reliability analysis was run on the dependent variable utilizing Cronbach's alpha (1970). This is a measure of internal consistency in participants' responses. It was found that there was good reliability between respondent's responses, with an alpha of 0.7683.

Other survey questions provided insights into the workloads of respondents and their preferences related to VAMTA training. It was reported that 64.3% of respondents saw 0–24 patients in sick bay daily. A daily count of 25–49 sick-bay visits was reported by 28.6% of respondents, while 7.1% reported 50–74 visitors per day.

When asked how much time they would be willing to devote to training a software system to recognize their voice, 21.4% of respondents said that a training period of less than 1 h would be acceptable. According to 57.1% of respondents, a training period of 1–4 h would be acceptable, while 21.4% of respondents said that they would be willing to spend 4–8 h to train the system. To train themselves to use VAMTA hardware and software applications, 42.9% of survey respondents said

they would be willing to undergo 1–4 h of training, while 57.1% said that they would train for 4–8 h. All respondents agreed that a longer training period would be acceptable if it would guarantee a significant increase in voice recognition accuracy and reliability.

12.4 Conclusions and Reflections

The recent 2009 VAMTA study revealed findings related to the effectiveness of the survey instrument and VAMTA itself. As a result of the study, a VAMTA follow-up questionnaire has been proven to be a valid survey instrument. By examining end-user responses from completed surveys, analysts were able to measure multiple variables and determine that the TTF model and a smart data strategy are applicable to the VAMTA system and are well received by end users. The survey's effectiveness extends its potential use in future studies of VAMTA's performance, in preventive healthcare in a national setting.

Analysis of the actual end-user responses supplied during the perceptual study confirmed that the TTF model does apply to VAMTA. In study survey responses, the VAMTA system received high ratings in perceived usefulness and perceived ease of use. This suggests that VAMTA shows promise for medical applications.

The survey responses also revealed that utility and quality performance are the major factors affecting the management of information by VAMTA. In the future, end-users who want to improve the management of healthcare information through use of VAMTA will need to focus on utility and quality performance as measured by perceived usefulness and perceived ease of use of the VAMTA system.

In addition to these recent findings related to TTF and the utility and quality performance of VAMTA, the original 2004 study demonstrated ways in which the VAMTA system itself can be improved. For example, additional training with the application and corrections of misrecognition improved the overall accuracy rate of this product. Lee [30] has noted that the characteristics of female voice pose certain technical challenges as an output, and in our case study likewise we noticed a similar technical challenge when the female voice is used as an input. Still, we resist the tendency to take a monolithic approach. While the gender of the user appears to impact the performance of VAMTA there are several *other* factors that may have impacted our results. For example, Lee and Carli[10,30] argue that task factors should be considered when taking into account the impact of voice output. They point out that tasks in high noise environments are more difficult to accomplish than those in quiet surroundings.

The combined findings resulting from the two studies have laid the groundwork for further testing of VAMTA. Additional testing is necessary to determine the system's performance in an actual national medical setting and to define more clearly other variables that may affect the TTF model when applied in that setting.

Given that in the recent 2009 study, efforts were focused on defining the information technology construct for global preventive healthcare applications, limited work was done to define tasks and individual characteristics for preventive care. To complete this work, future research should focus on defining the task and individual characteristics constructs for the TTF model for measuring user evaluations of IT in preventive healthcare.

At the very minimum, the 2004 VAMTA survey established criteria for developing a lightweight, wearable, voice-interactive computer capable of capturing, storing, processing, and forwarding data to a server for retrieval by users. Though the 2004 prototype met many of these expectations, limitations in the state of voice-recognition technologies created challenges for training and user interface. In 2004, an internal army review indicated that commercial, off-the-shelf products could not provide simultaneous walk-around capability and accurate speech recognition in the shipboard environment. Consequently, the adaptations of the 2004 existing technology involved trade-offs between speech recognition capabilities and wearability. The processors in lightweight, wearable devices were not fast enough to process speech adequately. Larger processors added unwelcome weight to the device, and inspectors objected to the 3.5 pounds during the walk-around surveys. In addition, throat microphones (used to limit interference from background noise) also limit speech recognition. These microphones pick-up primarily guttural utterances, and thus may miss those sounds created primarily with the lips, or by women's higher voice ranges. Heavier necks also impeded the accuracy of throat microphones. For most purposes, an SR1 headset microphone (Plantronics Inc., Santa Cruz, CA) focused at the lips was adequate for the system tested under conditions of 70–90 decibels of background noise.

Accuracy of speech recognition also depended on the time a user committed to training the device to recognize his or her speech, and changes in voice quality due to environmental or physical conditions. Accuracy rates varied from 85 to 98% depending on the amount of time users took to train the software. Optimal training time appeared to be 1 h for Dragon Naturally Speaking software and 1 h for VAMTA software. In addition, it must be remembered that there are both design and technology flaws. In order to overcome the design flaw in the study that requires the current software to interpret utterances in the context of an entire sentence, users had to form complete utterances mentally before speaking for accurate recognition to be performed.

Despite the limitations in speech recognition technology, the VAMTA prototype conducted in 2004 was successful in reducing the time needed to complete inspections, supporting local reporting requirements, and enhancing command-level intelligence. *Attitudes* of the users toward the hands-free mobile device were favorable, despite these restrictions, as evidenced by the findings of the 2009 in depth study which demonstrated that VAMTA end users were confident that the present VAMTA system saves time and improves the quality of medical encounters in which physicians entered medical data into patients' records via voice.

References

1. Adams, D.A., R.R. Nelson, and P.A. Todd. 'Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology: A Replication,' *MIS Quarterly*, 16:2, 1992, pp. 227–247.
2. Akaike, H. "Factor Analysis and AIC," *Psychometrika*, 52, 1987, pp. 317–332.
3. Bartkova, K. and Jouvet, D. "On using units trained on foreign data for improved multiple accent speech recognition." *Speech Communication, Volume 49, Issues 10–11, October–November 2007, Pages 836–846*.
4. Baroudi, J.J., M.H. Olson, and B. Ives. "An Empirical Study of the Impact of User Involvement on System Usage and Information Satisfaction," *Communications of the ACM*, 29:3, 1986, pp. 232–238.
5. Bergman, R.L. "In Pursuit of the Computer-Based Patient Record," *Hospitals and Health Networks*, 67:18, 1997, pp. 43–48.
6. Biket, D., Miller, S., Schwartz, R., & Weischedel, R. (1997). Nimble: A high performance learning name finder. *Proceedings of the Fifth Conference on Applied Natural Language Processing Association for Computational Linguistics*, 194–201.
7. Benzeghiba, M. et al. Automatic speech recognition and speech variability: A review *Speech Communication, Volume 49, Issues 10–11, October–November 2007, Pages 763–786*.
8. Bozdogan, H. "Model Selection and Akaike's Information Criteria (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 1987, pp. 345–370.
9. Burd, S. *System Architecture*, Course Technology, Boston: Massachusetts (2003).
10. Carli, L. Gender and Social Influence, *Journal of Social Issues* 57, 725–741 (2001).
11. Cash, J. and BR. Konsynski. "IS Redraws Competitive Boundaries," *Harvard Business Review*, 1985, pp. 134–142.
12. Cooke, M. et al. "Monaural speech separation and recognition challenge" *Computer Speech & Language, In Press, Corrected Proof, Available online 27 March 2009*.
13. Cronbach, L.J. *Essentials of Psychological Testing*. New York: Harper and Row, 1970.
14. Davis, F.D. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly*, 13:3, 1989, pp. 319–341.
15. Dixon, P. et al. "Harnessing graphics processors for the fast computation of acoustic likelihoods in speech recognition." *Computer Speech & Language*; Oct2009, Vol. 23 Issue 4, p510–526, 17p.
16. Duncan, W.J., P.M. Ginter, and L.E. Swayne. *Strategic Management of Health Care Organizations*. Cambridge, MA: Blakwell, 1995.
17. Ellsasser, K., J. Nkobi, and C. Kohier. "Distributing Databases: A Model for Global, Shared Care," *Healthcare Informatics*, 1995, pp. 62–68.
18. Fiscus, J. G. (1997) A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). *Proceedings, 1997 IEEE Workshop on Automatic Speech Recognition and Speech*.
19. Flynn, R. and Jones, E. "Combined speech enhancement and auditory modeling for robust distributed speech recognition" *Speech Communication, Volume 50, Issue 10, October 2008, Pages 797–809*.
20. George J. A. and Rodger, J.A. *Smart Data*. Wiley Publishing: New York 2010.
21. Goodhue, D.L. "Understanding User Evaluations of Information Systems," *Management Science*, 41:12, 1995, pp. 1827–1844.
22. Goodhue, D.L. and R.L. Thompson. "Task-Technology Fit and Individual Performance," *MIS Quarterly*, 19:2, 1995, pp. 213–236.
23. Hagen, A. et al "Highly accurate children's speech recognition for interactive reading tutors using subword units" *Speech Communication, Volume 49, Issue 12, December 2007, Pages 861–873*.
24. Haque, S. et al. "Perceptual features for automatic speech recognition in noisy environments" *Speech Communication, Volume 51, Issue 1, January 2009, Pages 58–75*.
25. Henderson, J. "Plugging into Strategic Partnerships: The Critical IS Connection," *Sloan Management Review*, 1990, pp. 7–18.

26. Hermansen, L. A. & Pugh, W. M. (1996). Conceptual design of an expert system for planning afloat industrial hygiene surveys (Technical Report No. 96-5E). San Diego, CA: Naval Health Research Center.
27. Ingram, A. L. (1991). Report of potential applications of voice technology to armor training (Final Report: Sep 84-Mar 86). Cambridge, MA: Scientific Systems Inc.
28. Joint Commission on Accreditation of Hospital Organizations. Accreditation Manual for Hospitals, 2009.
29. Karat, Clare-Marie; Vergo, John; Nahamoo, DaVAMTA (2007), “Conversational Interface Technologies”, in Sears, Andrew; Jacko, Julie A., *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (Human Factors and Ergonomics), Lawrence Erlbaum Associates Inc.
30. Lee,E. Effects of “Gender” of the Computer on Informational Social Influence: The Moderating Role of Task Type, *International Journal of Human-Computer Studies* (2003).
31. Longest, B.B. Management Practices for the Health Professional. Norwalk, CT: Appleton and Lange, 1990, pp. 12–28.
32. McLaughlin, C. and A. Kaluzny. Continuous Quality Improvement in Health Care: Theory Implementation and Applications. Gaithersburg, MD: Aspen, 1994.
33. McTear, M. Spoken Dialogue Technology: Enabling the Conversational User Interface, *ACM Computing Surveys* 34(1), 90–169 (2002).
34. Moore, G.C. and I. Benbasat. “The Development of an Instrument to Measure the Perceived Characteristics of Adopting an Information Technology Innovation,” *Information Systems Research*, 2:3, 1991, pp. 192–222.
35. Nair, N. and Sreenivas, T. “Joint evaluation of multiple speech patterns for speech recognition and training” *Computer Speech & Language*, In Press, Corrected Proof, Available online 19 May 2009.
36. Neustein, Amy (2002) “Smart’ Call Centers: Building Natural Language Intelligence into Voice-Based Apps” *Speech Technology* 7 (4): 38–40.
37. Ow, P.S., Mi. Prietula, and W. I-Iso. “Configuration Knowledge-based Systems to Organizational Structures: Issues and Examples in Multiple Agent Support,” *Expert Systems in Economics, Banking and Management*. Amsterdam: North-Holland, pp. 309–318.
38. Rebman, C. et al., Speech Recognition in Human-Computer Interface, *Information & Management* 40, 509–519 (2003).
39. Robey, D. “User Attitudes and Management Information System Use, *Academy of Management Journal*, 22:3, 1979, pp. 527–538.
40. Rodger, J.A. “Management of Information Technology and Quality Performance in Health Care Departments,” Doctoral Dissertation, Southern Illinois University at Carbondale, 1997.
41. Rodger, J. A., Pendharkar, P. C., & Paper, D. J. (1999). Management of Information Technology and Quality Performance in Health Care Facilities. *International Journal of Applied Quality Management*, 2 (2), 251–269.
42. Rodger, J. A., Pendharkar, P. C. (2004) A Field Study of the Impact of Gender and User’s Technical Experience on the Performance of Voice Activated Medical Tracking Application, *International Journal of Human-Computer Studies* 60, Elsevier 529–544.
43. Rodger, J. A. & Pendharkar, P. C. (2007). A Field Study of Database Communication Issues Peculiar to Users of a Voice Activated Medical Tracking Application. *Decision Support Systems*, 43 (2), 168–180.
44. Scharenborg, O. “Reaching over the gap: A review of efforts to link human and automatic speech recognition research” *Speech Communication*, Volume 49, Issue 5, May 2007, Pages 336–347.
45. Siniscalchi, M. and Lee, C.H. “A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition” *Speech Communication*, Volume 51, Issue 11, November 2009, Pages 1139–1153.
46. Torres, M. et al. Multiresolution information measures applied to speech recognition *Physica A: Statistical Mechanics and its Applications*, Volume 385, Issue 1, 1 November 2007, Pages 319–332.

Chapter 13

“You’re as Sick as You Sound”: Using Computational Approaches for Modeling Speaker State to Gauge Illness and Recovery

Julia Hirschberg, Anna Hjalmarsson, and Noémie Elhadad

Abstract Recently, researchers in computer science and engineering have begun to explore the possibility of finding speech-based correlates of various medical conditions using automatic, computational methods. If such language cues can be identified and quantified automatically, this information can be used to support diagnosis and treatment of medical conditions in clinical settings and to further fundamental research in understanding cognition. This chapter reviews computational approaches that explore communicative patterns of patients who suffer from medical conditions such as depression, autism spectrum disorders, schizophrenia, and cancer. There are two main approaches discussed: research that explores features extracted from the acoustic signal and research that focuses on lexical and semantic features. We also present some applied research that uses computational methods to develop assistive technologies. In the final sections we discuss issues related to and the future of this emerging field of research.

Keywords Modeling speaker state using computational methods • Speech processing

- Medical disabilities • Depression • Suicide • Autism spectrum disorder
- Schizophrenia • Cancer • Aphasia • Acoustic signals • Lexical and semantic features
- Mapping language cues to medical conditions

13.1 Introduction

Many medical conditions (e.g., depression, autism spectrum disorders (ASD), schizophrenia, as well as cancer) affect the communication patterns of the individuals who suffer from them. Researchers in psychology and psycho-linguistics have a long tradition of studying the speech and language of patients who suffer from these conditions to identify cues, with the hope of leveraging these cues for both

J. Hirschberg (✉)
Professor, Department of Computer Science, Columbia University,
2960 Broadway, New York, NY 10027-6902, USA
e-mail: julia@cs.columbia.edu

diagnosis and treatment. Like other observational data based on patient behavior, clinicians follow rigorous training to elicit and analyze patients' speech. More recently, researchers in computer science and engineering have begun to explore the possibility of finding speech-based correlates of various medical conditions using automatic, computational methods. If cues to medical disorders can be quantified and detected automatically with some degree of success, then this information can be used in clinical situations. Thus, automatic methods can assist clinicians in not only in screening patients for conditions, but also in assessing the progress of ongoing treatment. Furthermore, automatic methods can provide cost- and time-effective general screening methods for disorders, such as ASD, which often go undiagnosed. Finally, they can also provide useful input for assistive technologies that can be used in clinical situations or made available to patients in the home.

In this chapter, we discuss some of these approaches and suggest possibilities for future computational research on mapping language cues to medical conditions and on describing assistive technologies being developed to make use of them.

13.2 Computational Approaches to Speaker State

Computational approaches to the study of language correlates of medical conditions have largely arisen from related work on computational modeling of emotional state. Numerous experiments have been conducted on the automatic classification of the classic emotions, such as anger, happiness, sadness; secondary emotions such as confidence or annoyance; or simply positive from negative emotions, from acoustic, prosodic, and lexical information [39, 50, 3, 9, 26, 32, 33, 1]. Motivation for these studies have come primarily from call center and Interactive Voice Response (IVR) applications, for which there is interest in distinguishing angry and frustrated callers from the rest, either to hand them off to a human attendant or to flag such conversations as problematic for off-line study [26, 32] (Devillers & Vidrascu 2006, Gupta & Rajput 2007). Other research has focused on assessment of students' emotional state in automatic tutoring systems (Liscombe et al 2005b) [1].

More recently, emotional speech researchers have expanded the range of phenomena of interest beyond studies of the classic emotions to include emotion-related states, such as deception (Hirschberg et al. 2005), sarcasm (Tepperman et al. 2006), charisma (Biadsy et al 2008), personality [36], romantic interest [45], "hotspots" in meetings (Wrede & Shriberg 2003), and confusion (Kumar et al. 2006). To encompass this expansion of a research space which typically uses similar methods and a common set of features for classification, some have termed research of this larger class the study of *speaker state*. A recent focus of this area has been the use of techniques and features developed in studies of emotional speech in the analysis of medical and psychiatric conditions.

Most computational studies of emotion and other speaker state make use of statistical machine learning techniques such as Hidden Markov Models (HMMs),

logistic regression, rule-induction algorithms such as C4.5 or Ripper, or Support Vector Machines to distinguish among possible states. Corpus-based approaches typically examine acoustic and prosodic features, including pitch, intensity, and timing information (e.g., pause and turn durations and speaking rate), and voice quality, and less often lexical and syntactic information, extracted from large amounts of hand-labeled training data. Many corpus-based studies suffer from poor agreement among labelers, making the training data noisy. Since human annotation is expensive and labelers often disagree, unsupervised clustering methods are sometimes used to sort data into states automatically, but it is not always clear what the resulting clusters represent. Laboratory studies attempt to induce the desired states from professional actors or non-professional subjects in order to compare the same linguistic features in production studies or to elicit subject judgments of acted or natural emotions in perception studies. However, characteristics of emotions elicited from actors have been found to differ significantly from those evinced by ordinary subjects, making it unclear how best to design representative laboratory studies. More recently, Magnetic Resonance Imaging (MRI) studies have sought to localize various emotions and states within the brain (e.g., [23,27]). While these experiments sometimes produce intriguing results, it is still not clear what we can conclude from them, beyond the activation evidence in different locations of the brain associated with different speaker states. Thus, it is not always clear how best to study speaker state. Medical conditions, however, provide the possibility of correlating medical diagnoses with the same sorts of language-based features used to examine states like anger, confidence, and charisma.

13.3 Computational Approaches to Language Analysis in Medical Conditions

Computational approaches to the study of language in medical conditions can be classified in several ways – by the condition studied, the methods used, or the end goal of the study. Research has been done on prosodic cues for the assessment of coping strategies in breast cancer survivors [51], for evaluations of head and neck cancer patients (Maier et al. 2010), for diagnosis of depression and schizophrenia [2, 38, 40] (Bitouki et al 2009), and for the classification of ASD (ASD) [28, 46, 12, 51] (Hoque, 2008). Textual analysis methods, which rely on patient speech transcripts or texts authored by patients, have also been leveraged for cancer coping mechanisms [5,21], psychiatric diagnosis [42] (Elvevag et al. 2009), and analysis of suicide notes [43]. While much research has focused primarily on assessment, other researchers have explored possibilities for treatment, providing assistive technologies for those suffered from aphasia [14] or ASD [13]. Other work has examined the success of assistive technologies, such as the evaluation of cochlear implants [34]. In this section we will discuss some of this research to illustrate the approaches that have been taken and the current state of the field.

13.3.1 Assessment and Diagnosis

13.3.1.1 Cancer

Computational methods have been leveraged for diagnosing cancer based on a patient's speech transcript and quantifying the effect of cancer on speech intelligibility. One open research question in the field of psycho-social support for cancer patients and survivors is how to identify coping mechanisms. There has also been work on studying the presence and extent of emotion expressions in cancer patients' speech.

Oxman et al. [42] describe an early attempt at using automatic textual analysis to diagnose four possible conditions. The authors collected speech samples from 71 patients (25 with paranoid disorder, 17 with lung or breast cancer, 17 with somatization disorder, and 12 with major depression). Patients were asked to speak for 5 min about a subject of their choice. The speech transcripts were analyzed through two different textual analysis methods: (1) automatic word match against a dictionary of psychological dimensions [50] and (2) manual rating according to hostility and anxiety scales derived from the Gottschalk-Gleser scales [20]. The Gottschalk-Gleser scales are an established textual analysis method, traditionally used to support psychiatric diagnosis. The scales operate at the clause level, thereby taking into account a larger context than dictionaries. Two psychiatrists were also asked to read the transcripts and diagnose the patients with one of the four conditions. Neither the raters nor the two psychiatrists had knowledge of the patient's condition. The authors found that the pure lexical lookup method identified the best predictors for diagnosis classification, above the manual analysis and the expert diagnoses.

Automatic Speech Recognition (ASR) has been shown to be an effective means of evaluating intelligibility for patients suffering from cancer of the head and neck. In 2010, Maier et al. experimented with this method, recording German patients suffering from cancer of the larynx and others suffering from oral cancer. ASR was performed on the recordings; the word recognition rate (i.e. ratio of correctly recognized words to all words spoken by the speaker) was then compared to perceptual ratings by a panel of experts and to an age-matched control group. Both patient groups showed significantly lower word recognition rates than the control group. ASR yielded word recognition rates which correlated with experts' evaluation of intelligibility on a significant level. They thus concluded that word recognition rate from ASR can serve as a good means with low effort to objectify and quantify this important aspect of pathologic speech.

Zei Pollerman [51] found a positive correlation between patients considered to be coping well with their treatment and mean pitch range. It was hypothesized that *active coping* defined as "tonic readiness to act upon an event," could be reflected in the prosody of spontaneous speech. Ten breast cancer patients were diagnosed by clinicians as to their coping behavior, active or passive. Patients' voice recordings were recorded in high and low arousal conditions, and analyzed for mean f0, f0 range (defined as f0 maximum – f0 minimum), standard deviation of f0, mean intensity, intensity ratio expressed as decibel (dB) maximum vs. dB minimum, and

speaking rate. For each parameter the difference between values for high and low arousal conditions were measured. The study found that those with adaptive adjustment to their cancer (active coping) showed a higher difference in f0 range than those with passive coping behavior.

More recently, Graves et al [21] discovered some differences between cancer survivors and controls with respect to emotional expression in textual samples in a study of emotional expression in breast cancer patients. Comparing 25 breast cancer patients with 25 healthy patients, this study asked subjects to complete a verbal “emotion expression” behavioral task. James Pennebaker and his research collaborators’ Linguistic Inquiry and Word Count (LIWC) paradigm was used to identify positive and negative emotion words in text [41]. The authors found that, while there was no difference between cancer sufferers and healthy subjects, cancer patients used significantly fewer “inhibition words” and were in fact rated by trained raters as expressing *more* intense emotion.

The use of lexical resources to recognize expressions of emotion in text was also investigated in the work of Bantum and Owen [5]. They compare two automatic resources, LIWC and the Psychiatric Content Analysis and Diagnosis system (PCAD), based on the Gottschalk-Gleser scales mentioned above, for the recognition of positive and negative emotions, as well as more particular emotions of anxiety, anger, sadness, and optimism. The authors compiled a corpus of texts written by 63 women with breast cancer in an Internet discussion board. On average, each text contained 2,600 words. Trained raters annotated the texts according to positive and negative emotions, as well as presence of anxiety, anger, sadness, and optimism. Along with the texts, self-reports of emotional well-being were collected from the 63 participants. The authors found that LIWC was more accurate in identifying emotions than PCAD when compared to the manual raters, despite the context-sensitive nature of PCAD. Interestingly, when comparing the self reports to manual and automatic ratings, there was no significant correlation between the self-reported positive and negative emotions and the rater, LIWC or PCAD codes of positive and negative emotions.

13.3.1.2 Diabetes

Zei Pollerman [51] presents an early study of the potential use of acoustic-prosodic features in diagnosis of various conditions. In a study of diabetic patients at the University Hospitals of Geneva, the relationship between autonomic lesions and diminished emotional reaction was examined. About 40 diabetic patients’ autonomic functions were assessed by quantification of their heart rate variability (HRV). Emotional states (anger, joy, and sadness) were then induced via verbal recall of personal experience. Subjects were then asked to pronounce a short sentence in a manner appropriate to the emotion induced and to report the degree to which they had felt the emotion on a scale from 1 to 4. Their utterances were analyzed for f0, energy, and speaking rate and these features were then correlated with their HRV indices. The f0 ratio, that is, the difference between F0 maximum

and F0 minimum, energy range, and speaking rate were significantly correlated with HRV. A combined measure based on these features was then used to compare between subjects' productions of angry utterances and sad utterances. The study found that indeed subjects with a higher degree of autonomic responsiveness displayed a higher degree of differentiation between anger and sadness in their vocal productions. This suggested that poor prosodic differentiation between anger and sadness could be interpreted as a symptom of poor autonomic responsiveness. The study also found that groups with higher HRV reported a higher degree of subjective feeling for the induced emotions than those with lower HRV.

13.3.1.3 Depression

Researchers studying the acoustic correlates of depression have generally distinguished between studies of *automatic speech*, such as counting or reading, from studies of *free speech*, since the latter requires cognitive activity such as word finding and discourse planning in addition to simple motor activity. Research on automatic speech includes research at Georgia Tech and the Medical College of Georgia [38] on the use of features extracted from the glottal waveform to separate patients suffering from clinical depression from a control group. These researchers analyzed speech from a database of 15 male (6 patients, 9 controls) and 18 female (9 patients, 9 controls) recording reading a short story, with at least 3 min of speech for each subject; glottal features were used to classify within each gender group. While the data set was quite small, the researchers reported promising results for some of the features.

Other researchers have compared subjects diagnosed as *agitated* vs. those diagnosed as *retarded* depressives. Alpert et al. [2] examined acoustic indicators in the speech of patients diagnosed with depression to assess results of different treatments. In a 12-week double-blind treatment trial, that compared response to nortriptyline (25–100 mg/day) with sertraline (50–150 mg/day). Twelve male and ten female elderly depressed patients and an age-matched normal control group ($n=19$) were studied. Patients were divided into retarded or agitated groups on the basis of prior ratings. Measures of fluency (speech productivity and pausing) and prosody (emphasis and inflection) were examined. Depressed patients showed “less prosody” (emphasis and inflection) than the normal subjects. Improvement in the retarded group was reflected in briefer pauses but not longer utterances. There was a trend in the agitated group for improvement to be reflected in the utterance length but not the length of pauses. The authors concluded that clinical impressions were substantially related to acoustic parameters. These findings suggest that acoustic measures of the patient's speech may provide objective procedures to aid in the evaluation of depression.

Mundt et al. [40] presented a study in which they elicited, recorded and analyzed speech samples from depressed patients in order to identify voice acoustic patterns associated with depression severity and treatment response. An IVR telephone system was developed by Healthcare Technology Systems (Madison, WI) and used

to collect speech samples from 35 patients. All subjects got a personal pass and an access code to the IVR system and were then asked to repeatedly call a toll-free telephone number over a period of 6 weeks. The IVR system requested the subjects to respond to different types of questions such as “describe how you’ve been feeling physically during the past week” and additional tasks including to count from 1 to 20 or to recite the alphabet. The subjects’ depression severity was evaluated using three different clinical measures including clinician rated HAMDs, IVR HAMDs, and IVR QIDS. During the 6 weeks of data collection, 13 subjects showed a treatment response whereas 19 did not. Comparisons in vocal acoustic measurements between the group that showed treatment response and the group that did not were performed. The results show that there were no significant differences between subjects at baseline, that is, at the beginning of the 6 weeks. In a comparison of the acoustic measures between the baseline and the end of the 6 weeks, the group with no treatment response only showed a reduction in total pause time. In contrast, responders to treatment, showed a number of significant differences, including increased pitch variability (f_0), increased total recording sample duration, a reduction in total pause time, fewer number of pauses and an increase in speaking rate.

13.3.1.4 Schizophrenia

Schizophrenia is a neuro-developmental disorder with a genetic component. Patients typically show disorganized thinking and their language is correspondingly affected, especially at the discourse level. Research has found that healthy, non-schizophrenic relatives also exhibit some subtle peculiar communication patterns at the lexical and discourse level. Elvevag et al. (2007) show that Latent Semantic Analysis (LSA) is a promising method to evaluate patients based on their free-form verbalizations. In a follow-up study, Elvevag and colleagues (2009) collected 83 speech transcripts from three groups: schizophrenic patients, their first-degree relatives, and healthy unrelated individuals. They analyzed the transcripts according to three types of measures: statistical language models, measures based on the semantic, LSA-based similarity of a text sample to patient or control text samples, and surface features such as sentence length. They found that the three populations could be discriminated based on these three types of measures. When discriminating between patients and non-patients, surface features and language model features were predictive enough on their own (patients tended to have shorter sentences, with unusual choice of words). However, when discriminating between patients and their healthy relatives, a successful model required features related to syntactic and semantic level in addition to surface features.

Bitouki et al. (2009) have begun to examine the use of automatic emotion recognition approaches in speech to the diagnosis of schizophrenia. In their initial work, they have focused on identifying new features for emotion recognition. They have experimented with the use of segmental spectral features to capture information about expressivity and emotion by providing a more detailed description of the speech signal. They describe results of using Mel-Frequency Spectral Coefficients

computed over three phoneme type classes: stressed vowels, unstressed vowels, and consonants in the utterance to identify emotions in several available speech corpora, the Linguistic Data Consortium (LDC) Emotional Speech Corpus and the Berlin Emo-DB (Emotional Speech Corpus). Their experimental results indicate that both the richer set of spectral features and the differentiation between phoneme type classes improved performance on these corpora over more traditional acoustic and prosodic features. Classification accuracies were consistently higher for the new features compared to prosodic features or utterance-level spectral features. Combination of the phoneme class features with prosodic features leads to even further improvement. These features have yet, however, to be applied to the diagnosis of schizophrenia.

13.3.1.5 Autism Spectrum Disorders

Autism spectrum disorder is a range of neurodevelopment disorders that affect communication, social interaction and behavior. Symptoms range from mild to severe and include a lack of interest in social interaction, trouble communicating and repetitive and restrictive behavior. ASD has several diagnostic categories including autism, Asperger syndrome and Pervasive Developmental Disorder Not Otherwise Specified (PDD-NOS). Kanner [24,25] was one of the first to describe the behavioral disorders in the autistic spectrum and many of them are related to speech and language. Frequently mentioned language impairments include: unusual word choices, pronoun reversal, echolalia, incoherent discourse, unresponsiveness to questions, aberrant prosody, and lack of drive to communicate [48].

ASD has no simple confirmatory test, but is diagnosed by a set of physical and psychological assessments. Influenced by early observations made by Kanner [24] and Asperger [4], much work within psycholinguistics has been devoted to identifying and studying the language disorders related to ASD. The majority of this research has been qualitative rather than quantitative, but recently researchers have started to use computational methods to study some of these disorders.

ASD is associated with having an odd or peculiar sounding prosody [37]. Frequently mentioned deficits include both observations of a “flat” or monotonic voice as well as an abnormally large variation in f0. Deviations in prosody are difficult to isolate since prosody interacts with several levels of language such as phonetics, phonology, syntax, and pragmatics. Moreover, f0 varies a great deal between speakers, within speakers and over different contexts. An early study of f0 and autism suggests that compared with controls, a group of individuals with autism appeared to have either a wider or a narrower range of f0 [8]. [49] used the Prosody-Voice Screening Profile (PVSP), a standardized screening method, to study prosodic deficits in individuals with High-Functioning Autism (HFA) and Asperger Syndrome (AS). The results show that utterances spoken by the group of individuals with HFA and AS were often marked as inappropriate in terms of phrasing, stress and resonance.

Diehl et al. [12] use Praat [10] to extract f0 in order to explore if there are any differences in f0 range between individuals with HFA and typically developing children. The results show that the HFA individuals had a higher average standard deviation in f0 than controls; however, the groups did not differ in average f0. The clinical manual judgments, found by applying the standardized observational and behavioral coding metric known as Autism Diagnostic Observation Schedule (ADOS), of the individuals in the HFA group also turned out to be significantly correlated with the average standard deviation across f0 samples. That is, subjects with a higher variation in f0 were judged as having greater language impairment by trained clinicians. However, there is considerable overlap between the two groups, which suggests that f0 alone cannot be used to identify deviations in expressive prosody for ASD individuals.

There are also studies that have explored deviations in specific functions of prosody. Le Normand et al. [28] study prominence and prosodic contours in different types of speech acts. The speech samples were spontaneous speech taken from eight French speaking autistic children in a free play situation. The hypothesis was that children with a communicative disorder, such as autism will fail to produce appropriate prominence and prosodic contours related to different communicative intent such as declarative, exclamation or question. The speech acts and prominence were labeled manually. The prosodic contour was extracted and judged manually by visualizing the sound in the Praat editor [10]. The results suggest that there is a large proportion of the utterances with low prominence and flat prosodic contour.

Van Santen et al. [51] present another study that investigates how ASD individuals produce specific functions of prosody. The prosodic functions explored include lexical stress, focus, phrasing, pragmatic style, and affect. The tasks that were used to elicit data were specially designed to make the subjects produce speech with the targeted prosodic functions. The subjects were recorded and scored in real time by clinicians and later also judged by a set of naïve subjects. Automatically extracted measures of f0 and amplitude were collected as well. The results show that a combined set of the automatic measures correlated approximately as high with the naïve subjects’ mean scores as the clinicians’ individual judgments. However, the real time judgments of clinicians correlated substantially less with the mean scores than the automatic measures.

Paul et al. [46] investigate stress production by ASD individuals in a nonsense syllable imitation task. The aim was to establish whether ASD speakers produce stressed syllables differently from typically developing (TD) peers. The hypothesis was that the ASD patients would not perform differently from the control (the TD speakers). The study included speech samples from 20 TD speakers and 46 speakers with ASD. Subjective judgments and automatically extracted acoustic measures were correlated with diagnostic characteristics (e.g., PIQ, VIQ, Vineland and ADOS scores). The results show significant but small differences in the production of stressed and unstressed syllables between the ASD and TD speakers. First, the speakers with ASD were less likely to get the right subjective judgment of their produced stress than the TD speakers. Second, the analysis of the acoustic measures

revealed that both TD and ASD speakers produced longer stressed than unstressed syllables but that the duration differences between stressed and unstressed syllables were smaller for the ASD group.

Hoque (2008) analyzes a number of different voice parameters in individuals with ASD, Down syndrome (DS) and Neuro-Typical (NT). The parameters analyzed included f0, duration, pauses, rhythm, formants, and voice quality intensity. The parameters were explored using data mining methods in order to find a set of optimal features that can be used to identify distinguishable speech features for the ASD, DS, and NT groups. The results show that the average duration per turn was longer for NT than for ASD and DS. Moreover, the magnitudes of maximum rising and falling edges in a turn/utterance is much higher for NT than in DS and ASD. Yet, the number of rising and falling edges is comparable between NT and ASD. The future aim of this initial analysis of speech parameters is to build assistive technologies that can give individuals with ASD and DS real time feedback helping them produce more intelligible speech.

13.3.1.6 Suicide

There has been preliminary work on the analysis of suicide notes. The goal of such an analysis is to gain deeper understanding of the psychological state of individuals committing suicide, as well as to help prevent suicide of such individuals. Pestian et al. [45] envision a screening tool in place at a psychiatry emergency room that predicts the likelihood of an individual being in a suicidal state rather than merely depressed. The authors present preliminary results on the use of linguistic analysis when applied to suicide notes. Notes from 33 completers (individuals who completed suicide) and 33 simulators (individuals not contemplating suicide who were asked to write a suicide note) were collected. The authors trained a classifier for completer/simulator. Features included word count, presence of pronouns, unigrams, Kincaid readability index, and presence of emotional words based on an emotion dictionary match. The best classifier reached 79% accuracy in discriminating between completers and simulators. The most significant linguistic differences between completers and simulators were in fact at the surface level, such as word count. For comparison, five mental health professionals were asked to read the notes and classify them as originating from a completer or a simulator. Experts classified the notes with 71% accuracy.

13.3.2 Assistive Technologies

Much research has been done on the use of speech technologies to assist persons with medical disabilities, such as using Text-to-Speech systems as aids for the blind or for those who have lost their ability to speak. In this section however we focus on the use of recognition technologies to aid those who are being treated for disabilities.

13.3.2.1 ASD

Research at the MIT Media Lab led by Rosalind Picard has proposed a number of methods to assist those diagnosed with ASD. In El Kalioubi et al. [13] a wearable device is described which is designed to monitor social-emotional information in real time human interaction. Using a wearable camera and other sensors, and making use of various perception algorithms, the system records and analyzes the facial expressions and head movements of the person with whom the wearer is interacting. The system creators propose an application of individuals diagnosed with ASD, to help them in perceiving communication in social settings and enhancing their social communication skills.

Hoque et al (2008) analyzed the acoustic parameters of individuals diagnosed with ASD and Down syndrome. The idea is to use these parameters to visualize subject’s speech productions in real time in order to provide them with live feedback that can help them modify their productions. In further work [22], explores the effect of using an interactive game to help individuals with ASD produce intelligible speech. Nine subjects diagnosed with ASD and one subject with Down’s syndrome participated in the study. Most of the participants had difficulties with amplitude modulation and speech rate and the interactive game was designed to target these problems. The subjects alternately received sessions with a computerized game and traditional speech theory. A number of different acoustic measures were extracted automatically, including Relative Average Perturbation (RAP), Noise Harmonic Ratio (NHR), Voice Turbulence Index (VTI) and prosodic features including pitch, intensity, and speaking rate. A preliminary analysis suggests that one participant significantly slowed his speech rate when interacting with the computerized game. Furthermore, two other participants’ had significant reduction in pitch breaks when interacting with the computerized program, suggesting that they were able to better control their pitch.

13.3.2.2 Aphasia

Aphasia is a condition in which people lose some of their ability to use language due to an injury (often stroke) or disease that affects the language-production and perception areas of the brain. While aphasics are generally very motivated to improve their speech and language abilities and are receptive to using computer programs, they vary in their ability to use a mouse or keyboard, read, speak, and understand spoken language. Typically treatment currently includes speech therapy by trained therapists, which can be quite costly and is rarely covered by insurance. To address this problem, Fink et al [14,15] have developed software to provide aphasia sufferers with structured practice targeted at improving their speech on a long-term basis.

MossTalkWords 2.0 was developed by Moss Rehab Hospital (Philadelphia, PA) to lead users through several different types of exercises in a self-paced manner.

One of these exercises, Cued Naming, involves presenting a picture of an item or action and asking the user to name it, with cues available as memory aids. In the initial version of MossTalk, users self-monitored the correctness of their responses, or worked with a clinician. The need for the clinician could be reduced by using an ASR engine (Microsoft 6.1) with the grammar dynamically modified to include only the description of the picture being presented. An enhanced version of the system integrated speech recognition with MossTalkWords so that users would get immediate and automatic feedback from the ASR on whether the picture was correctly named. Advantages of using ASR instead of a human clinician are not only lower costs but also 24/7 availability of the system. An evaluation of the mild to moderate aphasia sufferers with good articulation found acceptable levels of accuracy for the ASR and considerable reported user satisfaction.

13.3.2.3 General Evaluation

Researchers at Universität Erlangen–Nürnberg have fielded a web-based system called Program for Evaluation and Analysis of all Kinds of Speech (PEAKSdisorders) to evaluate speech and voice disorders automatically. They particularly target speech evaluation after treatment, which is typically performed subjectively by speech pathologists, who are asked to assess intelligibility. The essence of their system is an ASR system developed at Erlangen–Nürnberg for use in spoken dialogue systems. The subject reads a known text, which is recognized by the system, which weights acoustic features higher than other components for this application. System output is just word error rate and word accuracy. This information is combined with information from a prosody module, which extracts pitch, energy, and duration along with jitter, shimmer, and information on voiced/unvoiced segments. These features are used to create a classifier trained on expert judgments. The resulting classifier is then used to assign scores to test patient recordings. Using their system, they report that their evaluation of patients whose larynx has been removed due to cancer and who have received tracheoesophageal (TE) substitute voices correlates 0.90 ($P < 0.001$) with human expert judgments. The correlation of PEAKS judgments with experts is 0.87 ($P < 0.001$) for children with cleft lip and palate who have undergone reconstructive surgery. The system can be accessed over the phone or on the web and is intended to provide a “second opinion” for pathologists working alone or in other cases where speech therapists might use additional information in further treatment.

13.4 Discussion

The use of computational methods to identify speaker state in the medical domain is an emergent field of research. This research builds on previous findings in the fields of psychiatry, psychology and cognition. Speech and textual analysis can help us gain a deeper understanding of medical conditions, and they can also contribute to the design of systems that can be used clinically, whether as an aid to

diagnosis/screening or to assess the effectiveness of treatment. While the methods described in this chapter are far from being readily usable in a clinical setting, they are nonetheless very promising, since they promise to help with many conditions which are currently very difficult to diagnose and treat.

13.4.1 Exploring Different Methods for Identifying Speaker State Identification

There are two main approaches to the analyses described above: one relies on the *speech signal* and one operates on the *speech content*. Historically, speech-processing researchers have focused on features derived from speech signal, while psychology researchers have relied on textual analysis methodology. More specifically, computational analyses of speech and language in the context of medical conditions operate at two primary levels:

1. Aspects of the speech signal, including durational features, intensity, and F0; and
2. Surface features of the textual, such as word count and lexical patterns, primarily examined through matching against lexical resources (see Pennebaker et al. [43] for a review of textual analysis methods and applications).

One open research question is whether combining these two levels, which have generally been investigated separately, could yield more accurate models of speaker state. Furthermore, as NLP technology progresses in part-of-speech tagging, syntactic parsing, semantic inference, and discourse modeling, more and more tools are now readily available and can be used in a variety of settings. It is worth investigating whether incorporating additional linguistic features yields better computational models of speaker state. So far, there is conflicting evidence that higher-level linguistic features are more helpful than shallow, lexical ones or speech signal information. In two studies, for instance, purely lexical features (as derived from lexical resources like LIWC) performed better than context-aware features, such as PCAD and the Gottschalk-Gleser scales [5,40]. On the other hand, Le Normand [28] shows that semantic and syntactic information derived from manually labeled speech acts can help target specific functions of prosody, which have been described previously as difficult for individuals with ASD to process. Similarly, in the study of schizophrenia, evidence shows that distributional semantics, without the necessity of factoring in speech signals, can be leveraged to discriminate patients from their first-degree healthy relatives (Elvevag et al. 2009).

13.4.2 Comparing Different Methods of Data Collection

One important characteristic shared by all the studies described in this chapter is their data-driven approach in characterizing speaker states. However, many open

questions about data collection remain, making it a primary concern for future research in this area.

One data collection issue, which needs to be carefully considered is the methodology used to elicit data. Many of the conditions and their associated speaker states have already been described in detail by clinicians in the literature (e.g., prosody in ASD patients). For computational methods to succeed, however, they must analyze actual speech samples which are representative of the behaviors under study and which can be easily segmented to target the representative examples. In the case of ASD patients, for instance, the goal is to collect speech samples which may exhibit particular prosodic patterns. In the study of coping mechanisms for cancer patients, speech samples with emotion expressions are desired.

13.4.2.1 Spontaneous Speech versus Scripted Speech

The research discussed in this chapter presents several strategies for collecting data:

1. Free spontaneous speech [28,42]
2. Free speech, albeit about a particular topic [5]
3. Proxy texts [45]
4. Specific tasks such as imitation, where subjects repeat words or sentence read to them [51].

Let us discuss imitation for a moment. Collecting data through specific tasks or repetitions facilitates segmentation, because it is possible to control what the subjects say and when they say it. Another advantage is that such tasks allow researchers to elicit larger amounts of the target behavior. Spontaneous speech, on the other hand, is difficult to analyze in a controlled fashion. The target behavior may occur sparsely, if at all. Moreover, in order to identify critical segments, spontaneous speech data requires hand labeling or other types of pre-processing, which can be challenging for research. Against all these advantages of scripted speech, spontaneous dialogue data has the benefit of *ecological validity* – it is more representative of subjects' behavior in a natural environment. For example, in the specially designed tasks presented by Van Santen et al. [51], the children appeared to have little problem conveying the target prosody. It is possible that ASD individuals can imitate appropriate prosody in a laboratory setting, but they may still have problems using these accurately in a real-world setting.

13.4.2.2 Annotation Difficulties and Shortcomings

In many cases, the speech samples, or their transcripts, must be annotated with gold standard information. For computational methods to be successful, one must pay attention to the annotation process. Because most annotations related to speaker state are largely observational, agreement among annotators can be low. Careful

annotation is often tedious and follows extensive annotation guidelines. For instance, annotating emotions in speech transcripts is a difficult task for humans. It requires trained annotators as well as established annotation schemas. Yet, while there are several emotion taxonomies developed in the field of psychology, it is unclear whether they are readily usable for computational purposes. While it makes sense from a psychological standpoint to differentiate between “anger” and “hostile anger” for example, it might be necessary to merge the two emotions when training an emotion-detection tool from annotated texts. Besides the annotation costs, the validity of the annotation is important. In several studies, expert opinions are not always reliable [42,45], but neither are patients’ self-reports [5].

13.4.2.3 Protecting Human Subjects’ Privacy

Finally, privacy concerns cannot be ignored when collecting language samples from patients. In the United States, for instance, the Health Insurance Portability and Accountability Act (HIPAA) and institutional review boards ensure that the privacy of patients is upheld. As such, health evaluations and also audio recordings are considered protected health information. Researchers must obtain institutional approval prior to collecting and processing speech samples. Furthermore, for datasets to be available to the scientific community, they must first be anonymized.

13.5 Conclusion

This is an exciting time for researchers in speech and language processing to investigate methods to recognize speaker state from a medical standpoint. With recent advances in speech processing, core natural language processing technologies, and data mining, the time is ripe to apply these methods to clinical applications. The resulting tools can impact medicine in several ways. Clinicians are more and more accustomed to having technology as part of their every-day activities and are more open to recognizing the value of technology in their decision-making processes. Thus, screening tools for conditions that are difficult to diagnose, partly because diagnosis of such conditions rely on close observation of patients over time, can be developed in tandem with the needs and skills of clinicians. Such tools can have economic and public health benefits, in that a wider population – particularly individuals who live far from major medical centers – can be efficiently screened for a broader spectrum of neurological disorders. Fundamental research on mental disorders, like post-partum depression and post traumatic stress disorder, and coping mechanisms for patients with chronic conditions, like cancer and degenerative arthritis, can likewise benefit from computational models of speaker state. A successful research endeavor, which brings together computational and clinical expertise, will ultimately provide better understanding of computational models as well as cognition.

References

1. H. Ai et al (2006), "Using System and User Performance Features to Improve Emotion Detection in Spoken Tutoring Dialogs," Interspeech 2006, Pittsburgh.
2. M. Alpert et al (2001), "Reflections of depression in acoustic measures of the patient's speech," *Journal of Affective Disorders*, 66:59–69.
3. J. Ang et al (2002), "Prosody-based automatic detection of annoyance and frustration in human-computer dialog", ICSLP 2002, Denver.
4. H. Asperger (1944) (tr. U. Frith (1991), "Autistic psychopathy in childhood," in U. Frith. *Autism and Asperger syndrome*. Cambridge University Press. pp. 37–92.
5. E. Bantum and J. Owen (2009), "Evaluating the Validity of Computerized Content Analysis Programs for Identification of Emotional Expression in Cancer Narratives," *Psychological Assessment*, 2009, 21(1): 79–88.
6. Emo-D B. Berlin Emotional Speech Corpus. (<http://pascal.kgw.tu-berlin.de/emodb/>).
7. D. Bitouk et al. (2009), "Improving Emotion Recognition using Class-Level Spectral Features," Interspeech 2009, Brighton.
8. C. Baltaxe (1984). "Use of contrastive stress in normal, aphasic, and autistic children," *Journal of Speech and Hearing Research*, 27:97–105.
9. A. Batliner et al, (2003) "How to find trouble in communication," *Speech Communication*, 40, pp. 117–143.
10. P. Boersma & D. Weenink (2005). PRAAT: Doing phonetics by computer (Version 4.3.14) [Computer program]. Retrieved from <http://www.praat.org>.
11. F. Burkhardt et al. (2005), "A Database of German Emotional Speech," Interspeech 2005, Lisbon.
12. J. Diehl et al (2009), "An acoustic analysis of prosody in high-functioning autism", *Applied Psycholinguistics*, 30(3).
13. R. el Kalioubi et al. (2006). "An Exploratory Social-Emotional Prosthetic for Autism Spectrum Disorders," in *Body Sensor Networks*. 2006. MIT Media Lab.
14. R.B Fink et al (2009). "Evaluating Speech Recognition in a Computerized Naming Program for Aphasia," American Speech-Language Hearing Association Conference. New Orleans, November.
15. R. B. Fink et al. (2002). "A computer implemented protocol for treatment of naming disorders: Evaluation of clinician-guided and partially self-guided instruction," *Aphasiology*, 16(10/11):1061–1086.
16. B. Elvevaag, P. Foltz, D. Weinberger, and T. Goldberg (2007), "Quantifying Incoherence in Speech: an Automated Methodology and Novel Application to Schizophrenia," *Schizophrenia Research*, 93:304–316.
17. B. Elvevaag, P. Foltz, M Rosenstein, and L. DeLisi (2009), "An automated method to analyze language use in patients with schizophrenia and their first degree-relatives," *Journal of Neurolinguistics*.
18. W. Goldfarb et al. (1972), "Speech and language faults in schizophrenic children. Journal of Autism and Childhood Schizophrenia, 2(3):219–233, 1972.
19. P. Gupta & N. Rajput, (2006), "Two-Stream Emotion Recognition For Call Center Monitoring", Interspeech 2006, Pittsburgh.
20. Gottschalk, L., Winget, C., & Gleser, G. (1969). Manual of instructions for using the Gottschalk-Gleser content analysis scales: Anxiety, hostility, and social alienation-personal disorganization. Berkeley: University of California Press.
21. K. Graves et al. (2005), "Emotional expression and emotional recognition in breast cancer survivors: A controlled comparison," *Psychology and Health*, 20:579–595.
22. M. E. Hoque et al. (2009), "Exploring Speech Therapy Games with Children on the Autism Spectrum," Interspeech 2009, Brighton.
23. T. Johnstone et al (2006), "The voice of emotion: an fMRI study of neural responses to angry and happy vocal expressions," *Social, Cognitive and Affective Neuroscience*, 1(3), 242–249.

24. L. Kanner (1946), “Irrelevant and metaphorical language in early infantile autism,” *American Journal of Psychiatry*, 103:242–246.
25. L. Kanner (1948), “Autistic Disturbances of Affective Contact,” *Nervous Child*, 2:217–2520.
26. C. M. Lee and S. Narayanan (2004), “Towards detecting emotions in spoken dialogs,” *IEEE Transactions on Speech and Audio Processing*, 2004.
27. S. Lee et al (2006), “A Study of Emotional Speech Articulation using a Fast Magnetic Resonance Imaging Technique,” Interspeech 2006, Pittsburgh.
28. M. Le Normand et al (2008), “Prosodic disturbances in autistic children speaking French,” *Speech Prosody*, Campinas, Brazil.
29. M. Lehtinen (2008), “The prosodic and nonverbal deficiencies of French- and Finnish-speaking persons with Asperger Syndrome,” Proceedings of the ISCA Workshop on Experimental Linguistics, Athens.
30. M. Levit et al (2001), “Use of prosodic speech characteristics for automated detection of alcohol intoxication,” ISCA Workshop on Prosody in Speech Recognition and Understanding, Red Bank NJ.
31. Linguistic Data Consortium, “Emotional prosody speech and transcripts,” LDC Catalog No.: LDC2002S28, University of Pennsylvania.
32. J. Liscombe et al (2005), “Using Context to Improve Emotion Detection in Spoken Dialog Systems,” Interspeech 2005, Lisbon.
33. J. Liscombe et al (2006), “Detecting Certainty in Spoken Tutorial Dialogues,” Interspeech 2006, Pittsburgh.
34. X. Luo et al (2006), “Vocal Emotion Recognition with Cochlear Implants,” Interspeech 2006, Pittsburgh.
35. A. Maier, T. Haderlein, U. Eysholdt, F. Rosanowski, A. Batliner, M. Schuster, E. Nöth (2009), “PEAKS – A systems for the automatic evaluation of voice and speech disorders,” *Speech Communication* 51 (2009):425–437.
36. F. Mairesse and M. Walker (2006), “Automatic Recognition of Personality in Conversation,” HLT-NAACL 2006, New York City.
37. G. Mesibov (1992). “Treatment issues with high-functioning adolescents and adults with autism,” In E. Schopler & G. Mesibov (Eds.), *High-functioning individuals with autism* (pp. 143–156). New York: Plenum Press.
38. Elliot Moore II, Mark Clements, John Peifer and Lydia Weisser (2003), “Investigating the Role of Glottal Features in Classifying Clinical Depression,” IEEE EMBS, Cancun.
39. S. Mozziconacci and D. J. Hermes (1999), “Role of intonation patterns in conveying emotion in speech,” ICPhS 1999, San Francisco.
40. Mundt, J. et al (2007), “Voice acoustic measures of depression severity and treatment response collected via interactive voice response (IVR) technology,” *Journal of Neurolinguistics*, 20(1):50–64.
41. P. Oudeyer (2002), “Novel useful features and algorithms for the recognition of emotions in human speech,” *Speech Prosody 2002*, Aix-en-Provence.
42. T. Oxman, S Rosenberg, P. Schurr, and G. Tucker (1988), “Diagnostic Classification Through Content Analysis of Patient Speech,” *American Journal of Psychiatry*. 1988. 145:464–468.
43. Pennebaker, J. et al (2001), *Linguistic Inquiry and Word Count: LIWC 2001*. Mahwah, NJ: Erlbaum.
44. J. Pennebaker, M. Mehl, and K. Niederhoffer (2003), “Psychological Aspects of Natural Language Use: our Words, our Selves,” *Annu. Rev. Psychol.* 2003. 54:547–77.
45. J. Pestian, P. Matykievicz, J. Grupp-Phelan, S. Arszman Lavanier, J. Combs, and R. Kowatch (2008), “Using Natural Language Processing to Classify Suicide Notes,” ACL BioNLP Workshop, pp. 96–97.
46. Paul, R et al (2008) ‘Production of syllable stress in speakers with autism spectrum disorders,’ *Research in Autism Spectrum Disorders*, 2:110–124.
47. R. Ranganath, D. Jurafsky, and D. McFarland (2009), “It’s Not You, it’s Me: Detecting Flirting and its Misception in Speed-Dates,” EMNLP 2009, Singapore.

48. Rapin, I., and Dunna, M. (2003), "Update on the language disorders of individuals on the autistic spectrum," *Brain Development*. 25:166–172.
49. Shriberg, L. et al. (2001), "Speech and prosody characteristics of adolescents and adults with high-functioning autism and Asperger syndrome," *Journal of Speech, Language, and Hearing Research*; 44(5).
50. P. Stone, D. Dunphy, M. Smith, et al (1969), "The General Inquirer: A Computer Approach to Content Analysis," Cambridge, Mass. MIT Press.
51. van Santen, J. et al (2009), "Automated assessment of prosody production," *Speech Communication* 51:1082–1097.
52. J. Yuan et al (2002), "The acoustic realization of anger, fear, joy, and sadness in Chinese," ICSLP, Denver.
53. Zei Pollerman, B. (2002), "A Study of Emotional Speech Articulation using a Fast Magnetic Resonance Imaging Technique," Speech Prosody 2002, Aix-en-Provence.
54. E. Zetterholm (1999), "Emotional speech focusing on voice quality," FONETIK: The Swedish Phonetics Conference, Gothenburg.

Chapter 14

“Cry Baby”: Using Spectrographic Analysis to Assess Neonatal Health Status from an Infant’s Cry

Hemant A. Patil

Abstract Infant cry analysis is a multidisciplinary area of research incorporating pediatrics, neurology, physiology, engineering, developmental linguistics, and psychology. It has been proposed in the pediatric literature that the infant cry is a reflection of complex neurophysiologic functions and that analysis of the cry itself can be used to assess the status of the infant’s health. Given the diagnostic importance of infant cry, this chapter presents application of spectrographic analysis to the vocal sounds of an infant, comparing normal with abnormal infant cry. Drawing from a rich body of research on spectrographic analysis predominantly used for performance of speaker recognition, this chapter presents how such spectral features that are used to identify and verify speakers can be applied to assess the neonate’s health status, by comparing a normal to an abnormal cry. Ten distinct cry modes, viz., hyperphonation, dysphonation, inhalation, double harmonic break, trailing, vibration, weak vibration, flat, rising, and falling have been identified for normal infant cry and their spectrographic patterns were observed. This analysis was then extended to abnormal infant cry. It has been observed that the *double harmonic break* is more dominant for abnormal infant cry in cases of myalgia (muscular pain). The *inhalation* pattern is distinct for infants suffering from asthma or other respiratory ailments such as a cough or cold. For example, for the infant whose larynx is not well developed, the *pitch harmonics* are nearly absent. As such, there are no voicing or glottal vibrations in the cry signal. In addition, for infants with Hypoxic Ischemic Encephalopathy (HIE), there is an initial tendency of pitch harmonics to rise and then to be followed by a blurring of such harmonics. Finally, an infant cry classification system is analyzed by observing the nature of the optimal warping path in the Dynamic Time Warping (DTW) algorithm.

Keywords Infant cry • Spectrographic analysis • Spectral features used for speaker identification and verification • Acoustic characteristics of normal vs. abnormal infant cries • Baby cry analyzers • Asthma • Myalgia • Larynx not developed

H.A. Patil (✉)

Assistant Professor, Dhirubhai Ambani Institute of Information and Communication Technology,
DA-IICT, Gandhinagar, Gujarat- 382 007, India
e-mail: hemant_patil@daiict.ac.in

(Laryngomalacia) • Hypoxic Ischemic Encephalopathy (HIE) • Dynamic time warping (DTW) • Pitch harmonics

14.1 Introduction

Speech is the most powerful means of communication for human beings. We can easily express our thoughts, emotions, ideas, etc. through speech. However, infants can communicate with us primarily by means of their cry. Infant cry is a sequence of motor performances and associated acoustic manifestation including vocalization, constrictive silence, coughing, choking, interruptions or various combinations of such performance [31]. Based on the cry and the environment, the parents or guardians empirically estimate the reason for the distress or may identify an infant from their cry [29]. Infant cry carries many levels of information such as emotions, health, gender, disease (abnormalities), preterm vs. full term, first cry, identity, etc. (as shown in Fig 14.1). For example, first cry of an infant is considered to be one of the factors for determining *Apgar count*, a measure to classify healthy vs. unhealthy or weaker newborns [1,34]. In addition, weight of the newborn could also be related to the cry [3]. Recently, the author has reported on an interesting experiment concerning identification and authentication of infants from their cry [29]. The main objective of this chapter is to investigate, using spectrographic analysis, the differences in acoustic characteristics of normal vs. abnormal infant cries. The study presented in this work may have its *social relevance* to investigate the causes for Sudden Infant Death Syndrome (SIDS).¹

¹SIDS-is a syndrome marked by the sudden death of an infant that is unexpected by history and remains unexplained after a thorough forensic autopsy, a detailed crime scene investigation, and an exploration of the medical history of the infant and family. SIDS was responsible for 0.543 deaths per 1,000 live births in the U.S. in 2005 [49,36,39,41]. According to a recent study, babies who die of SIDS have abnormalities in brain stem (the medulla oblongata) which helps in control functions like breathing (*which in turn may affect infant cry*), blood pressure, arousal and abnormalities in serotonin signaling. According to the National Institute of Health (NIH), which funded the study, this finding is the strongest evidence to date that the structural difference in a specific part of the brain may contribute to the risk of SIDS [30,49]. Colton and Steinschneider analyzed cry of SIDS victim and correlated the cause of SIDS with relatively lower *Fo*, longer duration, lower formant frequencies and greater sound pressure level throughout the spectrum [6]. Corwin *et. al.* observed that infants whose first cries exhibited a high first formant were more likely to die of SIDS than infants whose first cries did not have this characteristics [5]. In a groundbreaking study of 74 larynges removed at death from children who died of SIDS reported by Harrison, it was observed that SIDS can be attributed to reduction in the subglottic area (particularly around the age of 3 months), which is potentially lethal. The reduction in subglottic airway is often secondary to an increase in mucus-secreting glands caused by an upper respiratory tract infection, all of which affects infant cry [15]. These pioneering studies indicate that the infants with SIDS have significant abnormalities in their cry signal and hence changes in spectral characteristics. Moreover, imagine a situation where an infant cry analyzer is developed to increase doctors' confidence in making decisions about slow developing abnormalities in infants from their cry beforehand. In such cases, suitable steps in the form of *drugs or therapy* could be given to save the life of infant. In addition, the work on cry analysis of infants whose larynx is not fully developed (e.g. laryngomalacia) is scant. This present work may constitute a crucial step in filling this gap.

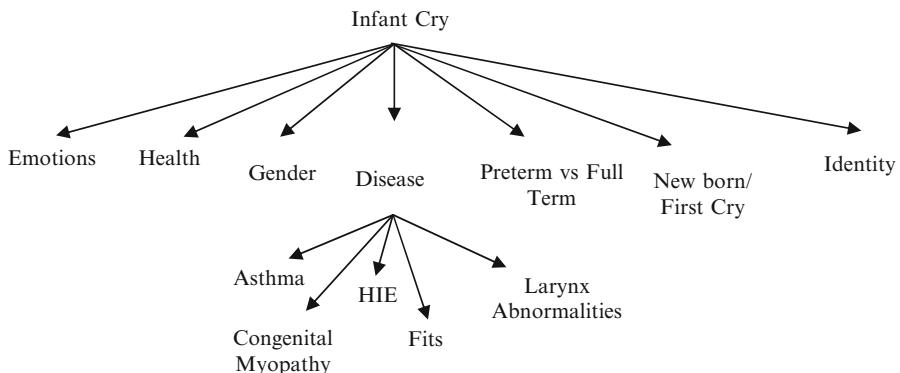


Fig. 14.1 Different levels of information conveyed in infant cry

The pioneering work on study of infant cry was started by Wasz-Hockert *et al.* in the 1960's in Scandinavia [44]–[48]. Since then, infant cry research has drawn great attention from various disciplines such as neurology, physiology, pediatrics, developmental psychology, comparative psychology, psychiatry, developmental linguistics, and engineering [51]. It is believed that infant cry research can unveil the complex psychological relationships and interactions between the care giving environment and the infant and can provide important information about the infants' anatomic and linguistic development [24–26,51]. Most of the earlier methods of infant cry analysis used features such as the latency, duration, fundamental frequency, and formant frequencies. This is due to the ease with which we can read these features from spectrograms. However, the modern signal processing techniques helped us to probe further into various perspectives in our field in order to facilitate our understanding of the physiological and anatomical basis of cry production [51]. For example, Prescott reported that the pitch contours and stop patterns are features of infant cry, which have been shown to have promising clinical applications [31]. A physioacoustic model of the infant cry is discussed in [13]. The work reported by Fort *et al.* uses parametric and non-parametric approaches such as linear prediction (LP) and cepstrum analysis for estimating the formant contour in the infant cry [17].

Spectrographic analysis finds its classic use in speaker recognition after the publication of an article by L. G. Kersta in Nature [19]. This was the first study on speaker identification based on *subjective* experiments (the subjects used in this study were eight female high school students 16–17 years old. They were given about one week of training in spectrogram reading and success rates were found from a common decision made by a panel of two girls). Even today, spectral features are dominantly used in speaker recognition [54]. In the infant cry analysis literature, spectrographic analysis is exploited for its characterization of the glottal source, i.e., pitch and its harmonic structure.

For example, Golub and Corwin reported that the sound source of infant cries is at the larynx and that three different modes of vibration exists, viz., full vibration at approximately 250–700 Hz (phonation), a falsetto-like vibration at about 1,000–2,000 Hz

(which may involve a thin portion of ligament only) (hyperphonation) and an aperiodic *turbulence* movement of the vocal folds (dysphonation) [13]. Xie *et al.* used these three basic cry modes to refine them into ten subtypes of infant cry modes, i.e., phonation is divided into five subtypes such as vibration, weak vibration, flat, rising, and falling (these last five related to pitch harmonics variations). In addition, they have defined two new cry modes, viz., inhalation (breathing) and double harmonic break (muscle tension) and these ten cry modes are used for automatic assessment of infant's level-of-distress from the cry signals using signal processing techniques such as cepstrum analysis, vector quantization and hidden Markov models [51,52]. These studies are, however, limited. They report on the *physical and or emotional situation* of normal (clinically healthy) infant cries and do not tackle the more demanding problem of performing close *comparative* analysis of normal vs. abnormal infant cries.

This chapter addresses this difficult task, showing the present use of these ten cry modes for comparative analysis of normal and abnormal infant cries with the objective of exploring possible techniques for classification. Infants suffering from asthma, larynx not developed (laryngomalacia) and Hypoxic Ischemic Encephalopathy (HIE), are classified as, for my own study purposes, clinically abnormal cases. In addition, use of the dynamic time warping (DTW) based approach is presented to investigate the class separability of different cry modes through the nature of the optimal warping path in DTW algorithm. Other significant studies in infant cry analysis literature are reported in [2,4,7,9–12,14,16,18,20,28,32,38,40,43,50,53,55,56].

14.2 Data Collection and Corpus Design

In this section, the details of experimental setup, data collection and corpus design for infant cry classification are presented. Cries of 184 infants were recorded at the following hospitals, viz., King George Hospital (K.G.H), Prabha Nursing Home (PNH), Child Clinic, Visakhapatnam, India. The duration of recorded infant cry varied from 15 to 40 s. The details of corpus are given in Table 14.1 [3]. The recording is done with a portable Cenix digital voice recorder. Same recording instrument was used at all places. During recording, the microphone was held

Table 14. I Details of infant cry database

| Item | Details |
|------------------------------------|---|
| No of infants | 184 |
| No. of sessions | 1 |
| Digital Recorder | VR-P2340, 64MB memory |
| Storage media | Embedded flash memory |
| Sampling Frequency | 12 kHz, PCM 16-bit resolution. |
| HQ | 500 Hz–5000 Hz |
| LQ | 500 Hz–3400 Hz |
| External microphone 3.5 Mono Jack. | This is an external input to The recorder |
| Acoustic environment | Hospital, Nursing Home (delivery ward) |

approximately 6–10 cm away from the infant’s mouth, so as to avoid any clipping of the sampled data. Due to the inherent limitations of collecting data from many infants, a portable device was used. All the sound files are stored in *wav* format, after being transferred to PC via an external PC Interface USB Cable 1.1. Details of the infants such as name, age, weight, ailments (if any), and the corresponding system that is affected, whether the child is being brought up by the parents themselves are noted. Some comments regarding the disease were also documented and stored in *text* file. And wherever possible, the commentary of the physician or doctor who diagnosed the infant was recorded and stored in *wav* file along with recording of the cries. Some of the experiences during data collection were as follows [3]:

1. Initially, the parents or the guardians of the infants were informed of the purpose of recording and their signatures regarding their consent for the same was taken. Fortunately, in all the cases, the presence and consent of the concerned doctor was crucial. Otherwise, the parents would not have understood the purpose of the study.
2. Almost 60% of the recordings were spontaneous cries. Patient monitoring of the infants was required for the same.
3. In the hospitals, the situation was such that while recording for one infant, another starts crying, this required swift movements from infant to infant. Besides, owing to the fact that infants cry mostly throughout nights and in the morning till around 10 am, recordings in the hospitals could be done only in the early hours of the days. Hence, there was a sort of virtual time constraint after which the infants would sleep.
4. Sometimes, while recording, the parents were pampering their children. Sometimes, even these were recorded.
5. There is an interesting correlation between the amplitude of cry and weight of the baby, i.e., cry of low birth weight infants is generally shrill, as compared to normal healthy babies. This can be subjectively attributed to the fact that sub-glottal system serves as a source of energy for the production of the sounds and since the infants with low birth weight naturally tend to have weaker breathes as compared to healthy heavier ones, their cries are shrill.
6. An interesting observation is that infants generally cry throughout the nights till around 10:00 am in the morning. Scientifically, the reason for this is that this sleeping pattern is synchronous with their sleeping pattern in mother’s womb. So most of the recording work in hospitals had to be done in the early hours of the day.

14.3 Lp Spectrum vs. Short-time Spectrum

In this section, related work on infant cry analysis using linear prediction (LP) spectrum and short-time spectrum is presented. In LP analysis, the combined effect of glottal pulse, vocal tract and radiation at lips can be modeled by a simple filter function $h(n)$, for a speech signal. A quasi-periodic impulse train is assumed for the voiced part and a random noise as the input for the unvoiced part at the output speech. The gain factor G accounts for the intensity (assuming a linear system).

Combining the glottal pulse, vocal tract and radiation yields a single all-pole transfer function given by

$$H(z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}}, \quad (14.1)$$

where $\{a_k\}_{k=1}^p$ are called as linear prediction coefficients (LPC) and they are computed by using *autocorrelation* method. From (1), difference equation for synthesizing the speech samples $S(n)$ is given by

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n).$$

The optimal values of $\{a_k\}_{k=1}^p$ are obtained by using the least square error (L^2 norm minimization) formulation of the linear prediction. This involves solutions of the normal equations given by

$$\sum_{k=1}^p a_k R(n-k) = -R(n); \quad n = 1, \dots, p,$$

Where

$$R(n) = \sum_{m=-\infty}^{+\infty} s(m)s(m-n)$$

is the autocorrelation function which constitutes one of the second order statistics [54]. The problem of linear prediction can be viewed from spectrum matching point of view as well. For example, given some signal spectrum say $S(\omega)$, we wish to model it by another approximate spectrum $\hat{S}(\omega)$ (through LP model) such that the integrated ratio between the two spectra, i.e., the total error is minimized. The total error is given by

$$E = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega; \quad \lim_{p \rightarrow \infty} \hat{P}(\omega) = P(\omega),$$

Where

$$P(\omega) = |S(\omega)|^2, \quad \hat{P}(\omega) = G^2 \left/ \left| 1 + \sum_{k=1}^p a_k e^{-jk\omega} \right|^2 \right.$$

and p is order of LP analysis. The limiting process in above equation says that we can approximate any spectrum arbitrarily closely by an all-pole model with respect to increase in LP order. For a given frame of infant cry, we can compute its Fourier spectrum by Hamming windowing of speech or infant cry frame followed by its magnitude of FFT. For computing LP spectra, first for fixed value of P LPC are computed along with the gain term in vocal tract filter by using autocorrelation method. Then LP spectrum can be computed by dividing G^2 by the magnitude

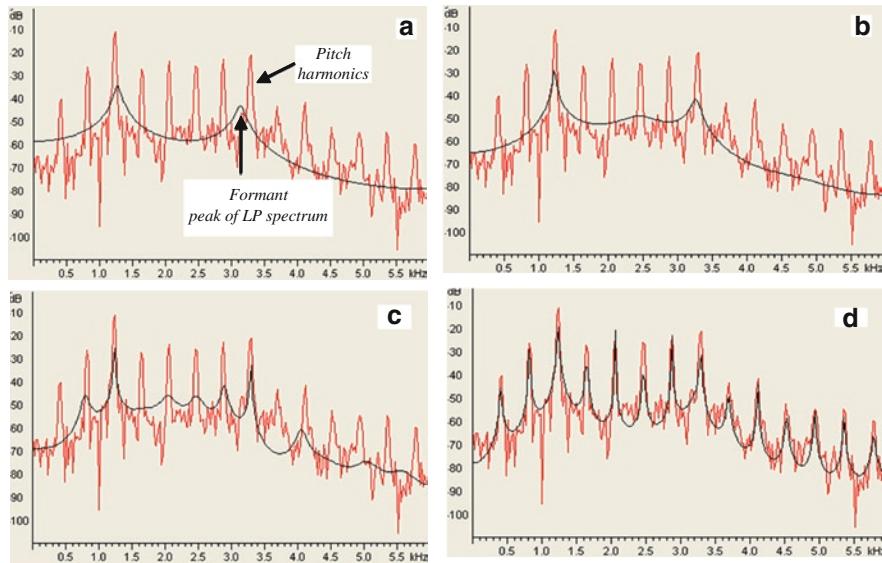


Fig. 14.2 Short-time spectrum and LP spectrum for voiced part of infant cry for different LP order p (a) $p=4$, (b) $p=12$, (c) $p=24$, (d) $p=34$ (After [29])

squared of the FFT of the sequence: $1, a_1, a_2, \dots, a_p$. Proper frequency resolution in LP spectrum can be obtained by simply appending appropriate number of zeros to this sequence before taking the FFT [23]. Figure 14.2 show short-time spectrum and corresponding superimposed LP spectrum for different LP orders ($p=4, 12, 24$ and 34). It is evident from the plots that, as the LP order is increasing from 4 to 34 , the peaks of LP spectrum try to match the pitch harmonics rather than formant structure of infant’s vocal tract (clearly evident from Fig 14.2d) and thus the spectral details in short-time spectrum are not captured very well by LP spectrum and hence possibly LP-derived *spectral* features may not be suitable for the present problem of study. Hence, short-time spectral features through spectrographic analysis are employed as the basis for my experiments in this work.

14.4 Sectrographic Analysis

As discussed in Sec. 14.1, spectrographic (or voiceprint) analysis finds its historic use in speaker recognition. According to Kersta, the solution to this problem is *feasible* in the sense that the parts which principally determine a speaker model (model describes *similar* acoustical characteristics of the speech of any given speaker) are the shape and size of the vocal cavities and the articulators. The vocal cavities are resonators which, much like organ pipes, cause energy to be reinforced in specific *spectrum* areas, depending on their sizes. The major cavities affecting

the speech are the throat, nasal, as well as two oral cavities formed in the mouth by the setting of the tongue. The contribution of the vocal cavities to voice uniqueness lies in their *size* and the *manner* in which they are coupled. These cavities have been approximately represented by two tubes, three tubes or four tubes models for nasals, fricatives and vowels. A still greater factor in determining the voice's uniqueness is the *manner* in which the articulators are manipulated during speech. The articulators include the lips, teeth, tongue, soft palate, and jaw muscles and the controlled dynamic interplay of these results in intelligible speech, which is not spontaneously acquired by an infant. It is a studied process by the infant through imitation of those around him who have mastered successful communication. The strong desire to communicate causes the infant to accomplish intelligible speech by successive steps of trial and error. Success requires that the infant learns a dynamic complex manipulation of interrelated muscles, controlling the movement of several articulators. Hence, the chance that the two individuals would have the identical dynamic use-patterns for their articulators would be remote. This makes us to believe that the two person's voices are unique, as reflected by the spectral energy distribution in their spectrograms, respectively [19].

In this section, an application of this spectrographic analysis is presented for assessment of normal vs. abnormal cry. Depending upon analysis window size, spectrograms are of two types, viz., wideband (window of length less than a pitch period) and narrowband (window of length equal to 2–3 pitch periods). By Heisenberg's uncertainty principle in signal processing framework, wider window widths (say rectangular) in time-domain creates narrower main-lobe width of *sinc* function in frequency-domain (Fourier transform of window) and hence we can distinguish pitch frequency and its harmonics clearly though *horizontal* striations in narrowband spectrograms. On the other hand, smaller window width creates overlapping of main-lobes of *sinc* function (called as spectral smearing) and hence it becomes difficult to distinguish pitch and its harmonics. With the source-filter model of speech production (cry is also a form of speech), we can express the narrowband spectrogram as a graphical display of the magnitude of the time-varying spectral characteristics, as is given by [57]

$$S(\omega, \tau) = |X(\omega, \tau)|^2 \approx \frac{1}{T^2} \sum_{k=-\infty}^{+\infty} |H(\omega_k)|^2 |W(\omega - \omega_k, \tau)|^2 \quad (14.2)$$

where $S(\omega, \tau)$ =spectral intensity as a function of frequency and center point of window, $w(n, \tau)$, width, T = pitch period of the infant cry signal, $H(\omega) = H(\omega)G(\omega)$, $H(\omega)$ =vocal-tract spectrum, $G(\omega)$ =spectrum of glottal flow waveform over single pitch period, $\omega_k = (2\pi / T)k$ and $(2\pi / T)$ =fundamental (pitch) frequency (in Hz).

From (2), it is clear that the term $|H(\omega_k)|^2$ must mirror the transfer function of the supralaryngeal vocal tract since they are spaced farther apart than the harmonics of the laryngeal excitation and at inharmonic intervals. However, according to the Lieberman *et al.* these energy concentrations may not exactly specify the formant frequencies of infant's vocal tract since harmonics of the laryngeal excitation

(spectrum of glottal flow over several pitch periods) are spaced normally at higher intervals (due to very high pitch) [21,22]. However, taking this uncertainty into account, we can still approximate the formant frequencies and hence infer the configuration of the infant’s supralaryngeal vocal tract for this vocalization by making use of Fant’s acoustic theory of speech production [8]. This theory allows us to infer that the supralaryngeal vocal tract configuration of this infant approximated as a 7.5-cm long uniform tube, which is open at one end. The resonances are given by [8,57]:

$$F_k = (2k + 1)c / 4l; k \geq 0$$

where c =velocity of sound, l =length of infant’s vocal tract. For $l=7.5$ cm, the first three formants of such a tube will occur at $F1=1.1$ kHz, $F2=3.3$ kHz, and $F3=5.5$ kHz [21,22]. Since pitch harmonics and their variations forms eight distinct cry modes out of ten, narrowband spectrograms are used in this work. Let us first follow definition of ten distinct cry modes from spectrograms of infant cry reported by Xie *et al.* [51] which was in turn motivated by earlier studies reported in [42,13].

1. *trailing (glottal roll)* – occurs at the end of long and powerful expiratory phonation. It is characterized by (a) a very low, gradually decreasing, and vibrating pitch frequency Fo and (b) a gradually decreasing total energy level.
2. *flat* – the basic expiratory phonation characterized by (a) a smooth and steady Fo (b) clearly observable harmonics, and (c) little energy distributions in the harmonics.
3. *falling* – similar to the *flat* except for a descending Fo .
4. *double harmonic break* – a simultaneous parallel series of harmonics in-between the harmonics of Fo the in-between harmonics occur suddenly and are usually weaker than the primary ones (may correlate well with abnormality).
5. *Dysphonation* – this is special feature (may also correlate well with abnormality) which results due to an aperiodic *turbulence* movement of the vocal ligaments and it is characterized by unstructured energy distribution over all the frequency range, sometimes with a tendency of higher concentration over the middle to high (1–5 kHz) frequency range. Sometimes, double harmonic break and dysphonation are correlated with some abnormality or muscle pain.
6. *Rising* –similar to flat except for an ascending Fo .
7. *Hyperphonation* – phonation with an extraordinarily high Fo (typically over 1 kHz)
8. *Inhalation* – the sound produced by the infant’s rapid breathing in of air. This usually occurs after an exhaustive expiratory phase.
9. *Vibration* – characterized by (a) clearly observable harmonics but with a vibrating Fo , (b) no unstructured energy distribution in between harmonics, and (c) a normally high total energy level.
10. *weak vibration* – similar to the vibration except that the total energy level is significantly lower than normal level.

Next, use of these cry modes for normal and abnormal infant cries through spectrographic (narrowband) analysis is presented:

14.4.1 Normal Infant Cry

The cry modes were extracted from different normal infants (less than eight months old) for various conditions associated with each infant during the recording task, such as before urinating, during urinating, extreme hunger, post injection cry, etc. Fig 14.3 shows the spectrograms of the ten cry modes. For each spectrogram, the frequency ranges from 0 Hz to 6 kHz (since sampling frequency=12 kHz). Figures 14.4–14.6 show occurrence of each of these cry mode in the neighborhood of the other. Arrows in these figures indicate zones of strong activity of a particular cry mode. In addition, the doctor's comments during recording of the infants are written in Box 14.1.

Box 14.1 Doctor's comments during recording of normal infant cry

Following comments are recorded after recording of each *normal* infant cry.

- 1) *Cry while passing urine* – “The child is about to pass urine, child cried indicating it is a pain for stimulus for a child (and the child expresses only by crying) and after passing urine, child stopped crying. This is a typical cry of a normal and healthy child.”
- 2) *Post injection cry*- “The cry may be also due to injection called as post injection cry.”
- 3) *Hunger cry* – “The baby is brought with persistent cry. History reveals that the mother is not able to produce adequate milk so the child is fed with artificial diluted milk. So this cry is probably because of severe hunger and after giving proper milk, the child has stopped crying. This is a classical hunger cry.”
- 4) *Cry due to wet diaper* – “This child is one month old. The child is about to pass motion and after passing the motion the diaper is still so wet the child doesn't like it and it has to be removed and until that the child keeps crying. This is absolutely normal cry in newborns.”
- 5) *Cry for upper respiratory tract infection (URT)* – “This is a 7 month old child with URT. It's reasonably normal cry.”

Some of the observations from Figs 14.4–14.6 are as follows:

1. Pitch harmonics are clearly visible (i.e., they are very strongly evident in spectrograms)
2. Pitch harmonics are constant for some time and then there is trailing
3. There is no frequent inhalation indicating no breathing difficulty
4. No sudden jump in rising or falling pattern, i.e., smooth transition
5. Double harmonic break and dysphonation of very “low strength” are observed some times.

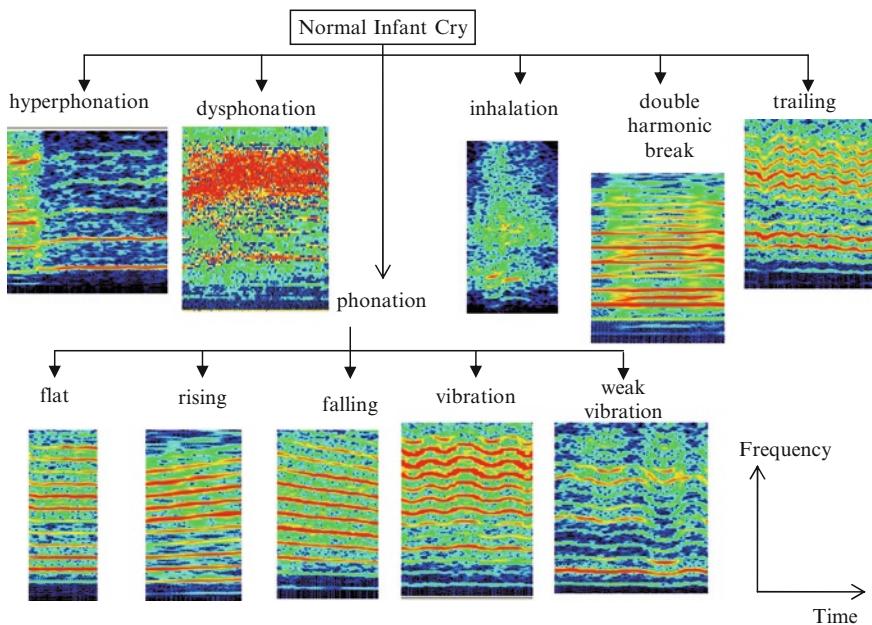


Fig. 14.3 Spectrographic analysis for normal infant cry

14.4.2 Infant Cry with Asthma

Asthma is the most chronic disease of childhood caused by airway reversible obstruction (due to airway inflammation – the hallmark of asthma – is probably initiated by immune system effects especially selective maturation of T-lymphocyte subtypes and allergic sensitization), which places substantial burden on the individual, the family and society. Variety of factors that contribute for asthma initiation are prenatal exposures, perinatal factors, breast-feeding and nutritional factors, childhood infections, specific allergies and indoor and outdoor air pollutants [37]. Figure 14.7 shows the spectrograms of the ten cry modes. In addition, the doctor’s comments during recording of the infants are written in Box 14.2.

Box 14.2 Doctor’s comments during recording of infant cry with asthma

Following comments are recorded after cry recordings of each infant with asthma

- 1) *Infant 1* – “This is a case of two year old child having asthma during treatment still not controlled and when he came back to the doctor, the child is crying. Still having asthma.”
- 2) *Infant 2* – “This is cry of a known asthmatic patient. Now comes with acute bronchiate asthma. This is an abnormal cry.”

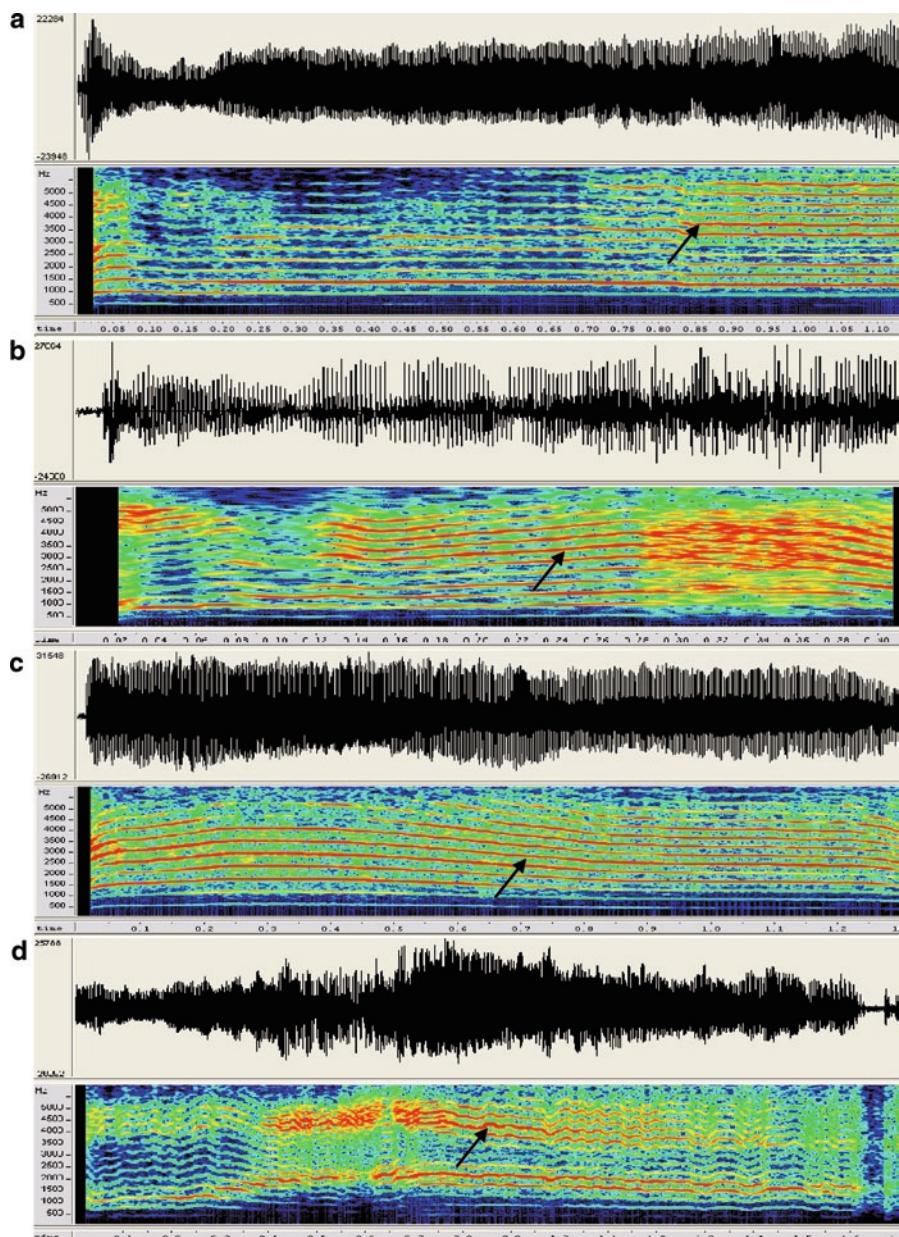


Fig. 14.4 Cry modes of normal infant cry: (a) flat, (b) rising, (c) falling and (d) trailing. Arrows indicates zone of strong activities of a particular cry mode

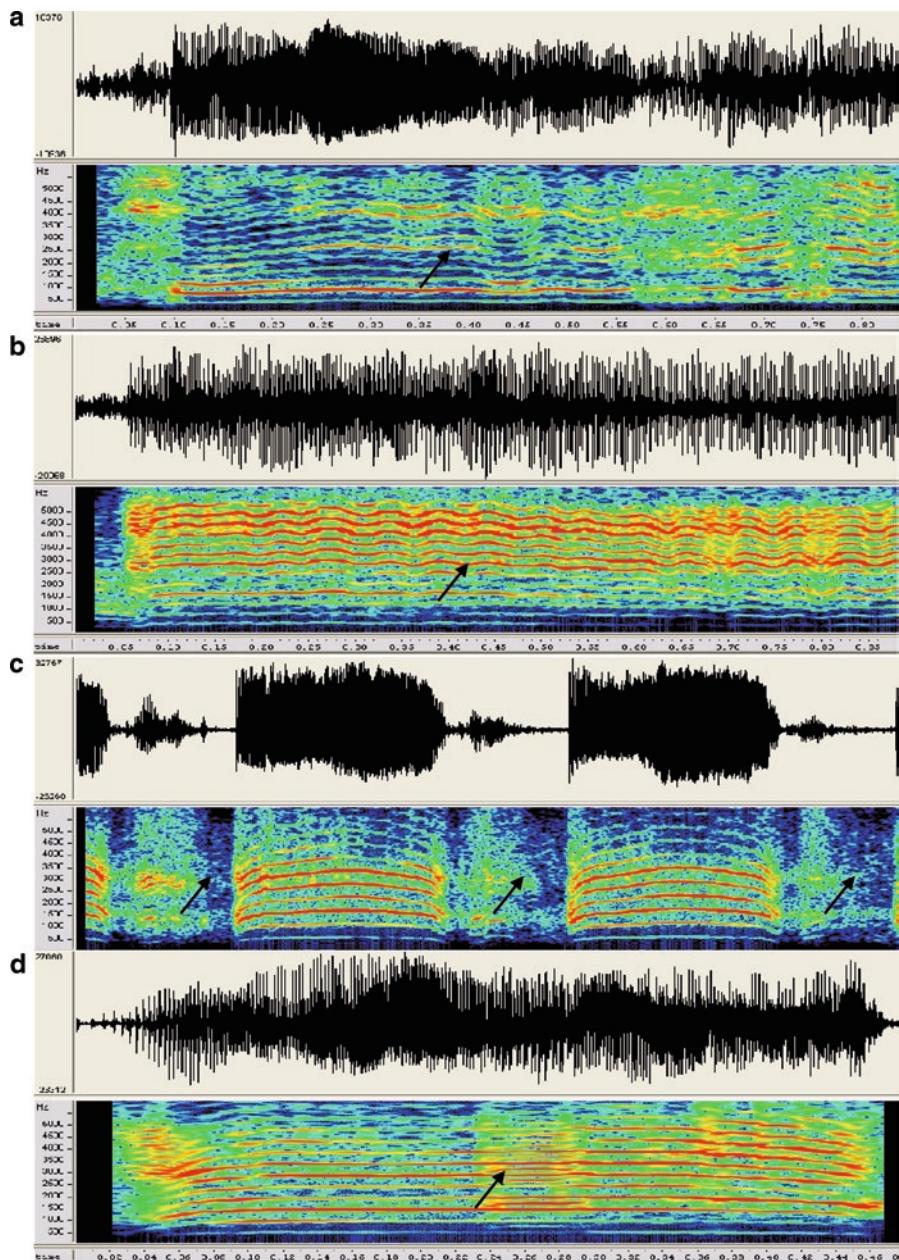


Fig. 14.5 Cry modes of normal infant cry (continued): (a) weak vibration, (b) vibration, (c) inhalation and (d) double harmonic break (weaker). Arrows indicates zone of strong activities of a particular cry mode

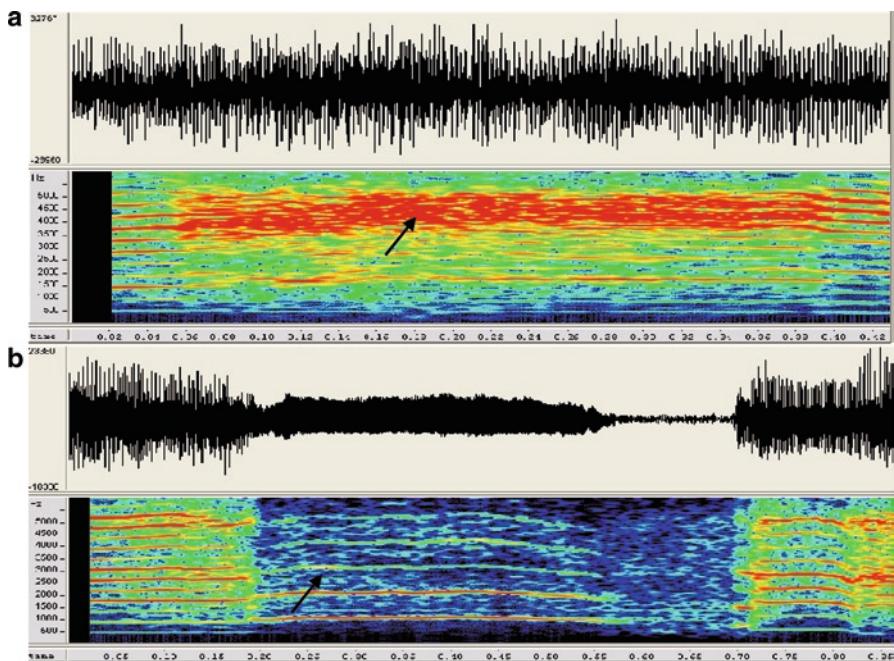


Fig. 14.6 Cry modes of normal infant cry (continued): (a) dysphonation and (b) hyperphonation. Arrows indicates zone of strong activities of a particular cry mode

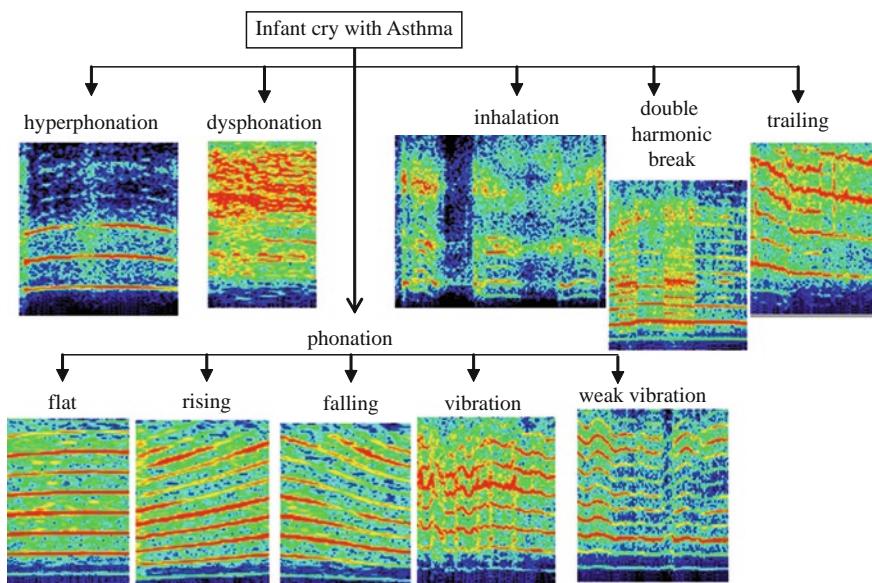


Fig. 14.7 Spectrographic analysis for infant cry with Asthma

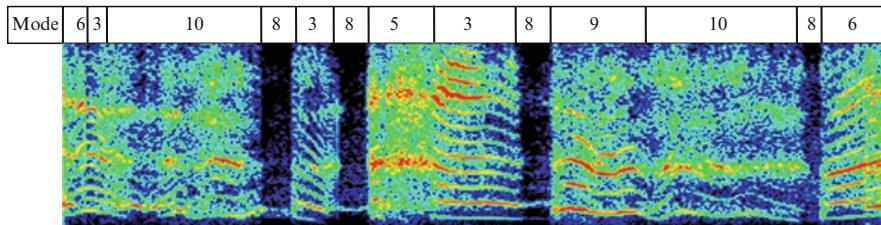


Fig. 14.8 A typical cry episode of the cry modes shown in Fig. 14.7. Typical cry modes are also labeled at the top

Figure 14.8 shows an example of labeling for cry mode of a particular cry episode of an infant suffering from asthma. On the top of the spectrogram shown in Fig 14.8, the segmentation or labeling of the cry signal is shown, and the number in an interval there indicates the type of cry mode (discussed at the beginning of this section) during that interval. For example, in Fig 14.8 first there is rising followed by falling; then weak vibration followed by inhalation and so on. Some of the observations from Figs 14.7 and 14.8 are as follows:

1. There is frequent inhalation indicating severe breathing difficulty
2. Spectral smearing (poor frequency resolution) of double harmonic break, vibration, weak vibration, trailing is observed.
3. Other cry modes, viz., flat, rising, falling, dysphonation, and hyperphonation seems to be less altered

14.4.3 Infant Cry with Larynx Not Developed (Laryngomalacia)

Laryngomalacia is a most common cause of congenital stridor and a kind of abnormality of the laryngeal cartilage. It may represent a delay of maturation of the supporting structures of the larynx (i.e. infants whose larynx is not developed at birth or shortly afterward). The infants with this abnormality produce a typical sound (chronic noisy breathing) which is a mostly unvoiced sound, i.e., the child is not able to produce glottal vibration (i.e., glottal activity [27]) inside a larynx.

Figure 14.9 shows the spectrograms of the ten cry modes (with some cry modes missing in this abnormal cry). In addition, the doctor’s comments during recording of the infants are written in Box 14.3. Figure 14.10 shows an example of labeling for cry mode of a particular cry episode of infant suffering from laryngomalacia. On the top of the spectrogram shown in Fig 14.10, the segmentation or labeling of the cry signal is shown, and the number in an interval there indicates the type of cry mode during that interval similar to Fig 14.8. For example, in this type of abnormality dysphonation and inhalation seem to dominate the entire cry episode with very little voicing.

Box 14.3 Doctor's comments during recording of infant larynx not developed (Laryngomalacia)

Following comments are recorded after cry recordings of infant with abnormalities in larynx

"This is a 4 month old child. This is actually not a cry. This is the case of congenital stridor where the larynx hasn't fully developed and the child has an inspiratory stridor. This sometimes happens when complicated by upper respiratory tract infection. It is very hazardous. This has to be treated early. This is typical case of stridor in infancy. The diagnosis for this cry or sound is infant suffering from congenital laryngeal stridor (Laryngomalacia). "

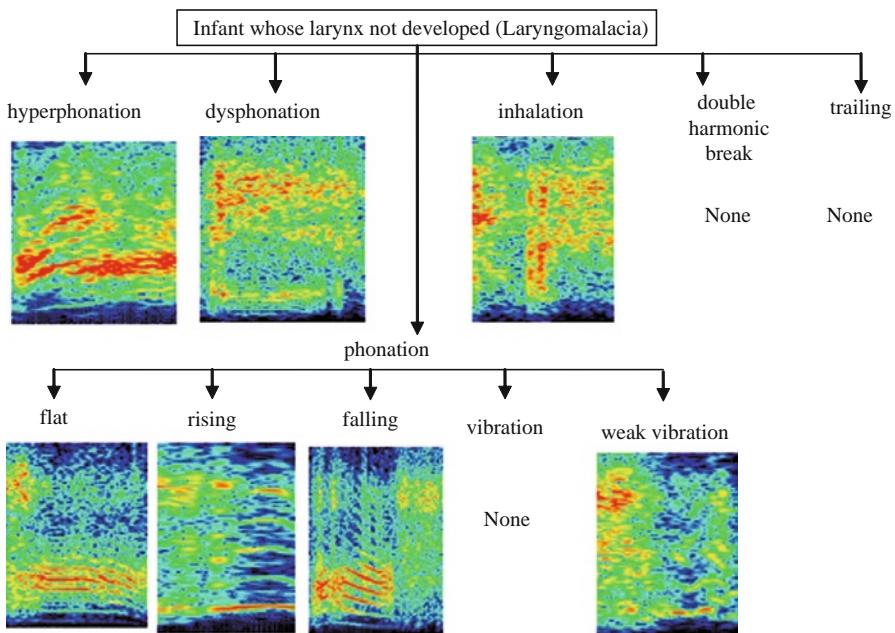


Fig. 14.9 Spectrographic analysis for infant whose larynx not developed (Laryngomalacia)

Some of the observations from Figs 14.9 and 14.10 are as follows:

1. There is frequent inhalation with different perceptual quality and energy indicating infants repeated attempts to make *glottal activity* (i.e., vibration of vocal folds, which is the primary mode of excitation of the vocal-tract system during speech production [27])
2. Spectral smearing (poor frequency resolution) over almost entire frequency range
3. Cry modes, viz., flat, rising, and falling, weak vibration are present with very weak energy harmonic structure indicating very mild or no glottal activity,

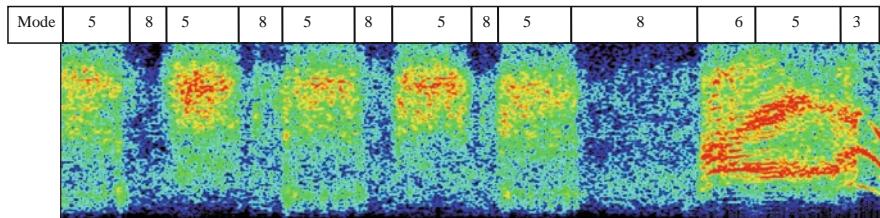


Fig. 14.10 A typical cry episode of the cry modes shown in Fig. 14.9. Typical cry modes are also labeled at the top

i.e., infant is not able to produce excitation to the vocal tract through the sudden closure of the glottis (i.e., localized impulse of high energy)

4. Other cry modes which are related to strong glottal activity, viz., double harmonic break, vibration, trailing are not at all observed in spectrograms

14.4.4 *Infant Cry with HIE (Hypoxic Ischemic Encephalopathy) or Asphyxias*

This is a disease caused to the newborn baby due to lack of supply of blood and oxygen to the brain. Due to this, the function of the human brain is disturbed and hence the neurophysiologic actions of the infant, to convey message of pain or of abnormalities through their cry, will be disturbed.

Figure 14.11 shows the spectrograms of the ten cry modes (with some cry modes missing in this abnormal cry). In addition, the doctor's comments during recording of the infants are written in Box 14.4. Figure 14.12 shows an example of labeling for cry mode of a particular cry episode of infant suffering from HIE. On the top of the spectrogram shown in Fig 14.12, the segmentation or labeling of the cry signal is shown, and the number in the interval there indicates the type of cry mode during that interval similar to Figs 14.8 and 14.10. For example, first there is inhalation cry mode followed by rising cry mode. Then there is sudden appearance of dysphonation cry mode followed by pitch falling cry mode and so on. In this type of infant cry, the spectral energy is not very dominant as these infants are poor criers and their cry is not vigorous (therefore less spectral energy). Some of the observations from Figs 14.11 and 14.12 are as follows:

Box 14.4 Doctor's comments during recording of infant cry with HIE

Following comments are recorded after cry recordings of each infant with HIE
 “This is a new born with second day with poor cry this is a case of HIE with seizures.”

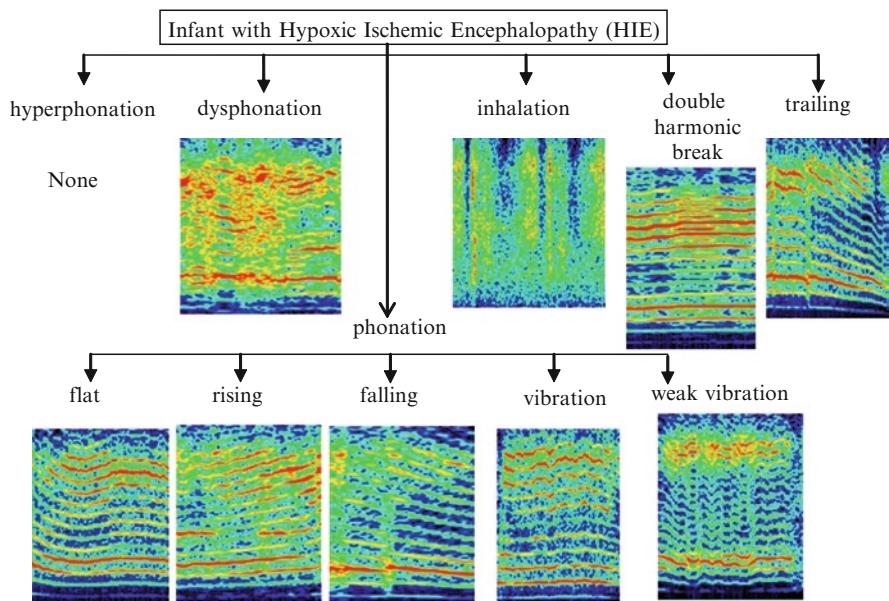


Fig. 14.11 Spectrographic analysis for infant suffering from HIE

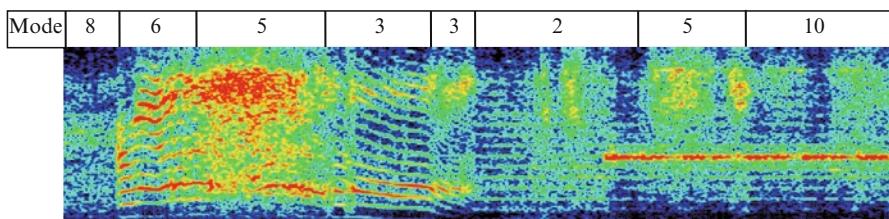


Fig. 14.12 A typical cry episode of the cry modes shown in Fig. 14.11. Typical cry modes are also labeled at the top

1. There is a tendency of pitch harmonics to rise followed by their blurring (i.e., *unstructured* spectral energy distribution) in the entire frequency range and then falling (cry mode) of the pitch harmonics. This sudden blurring of the pitch harmonics in between rise and falling cry modes may occur due to the fact that the infant is not able to vocalize (glottal activity) due to the inadequate supply of blood and oxygen to his brain. And hence may create disruption in adequate neural firing into brain to send signals to respective muscles of the larynx.
2. Hyperphonation is not present.
3. Overall spectral energy level seems to be low in entire frequency range.

14.5 Infant Cry Classification using DTW warping path

The main idea behind DTW was to exploit the non-linear time-normalization through dynamic programming (DP) so as to remove the nonlinear fluctuations in speech pattern due to speaking rate variation. In DTW, the time-axis fluctuation is approximately modeled with a nonlinear warping function of some carefully specified properties. Timing differences between two speech patterns are minimized by warping the time axis of one so that the maximum coincidence is attained with the other. Then, the time-normalized distance is calculated as the minimum residual distance between them [33], [35]. Recently, Yegnanarayana *et al.* have proposed an interesting approach of exploiting the nature of warping path in DTW algorithm to derive duration and pitch information for the text-dependent speaker verification task [54]. In this section, different cry modes are used through short-term spectral features of such modes (i.e., STFT magnitudes) so as to investigate the relative significance of these various cry modes for infant cry classification, by observing the nature of the optimal warping path of the DTW algorithm. For feature extraction, infant cry signal for each cry mode is divided into frames of 5.3 ms (64 samples for 12 kHz sampling frequency and which typically corresponds to 2–3 pitch periods) with 75% overlap. Hanning window is applied to each frame followed by computation of STFT magnitudes (known as feature vector). The cosine distance (inner product) between STFT magnitudes for a particular cry mode of normal (reference) and abnormal (test) infant cry is calculated to find the local match and then the dynamic programming algorithm is applied to find the optimal warping path whose computational details and algorithmic constraints are described in next paragraph.

Suppose there are two time series of feature vector matrices, A and B , having n and m number of feature vectors for test (abnormal cry) and reference (normal cry), respectively, i.e., $A = a_1, a_2, \dots, a_n$, and $B = b_1, b_2, \dots, b_m$ where each a_i and b_i are d -dimensional spectral feature vectors

To align these sequence of feature vectors (called as spectral trajectory in d -dimensional feature space) one constructs an n -by- m matrix where the (i th and j th) element of the matrix contains the distance $d(a_i, b_j)$ between the two points a_i and b_j (cosine distance in present case). A warping path W is a contiguous (in the sense stated below) set of matrix elements that defines a mapping between A and B . The k th element of W is defined as $w_k = (i, j)_k$ so we have:

$$w = w_1, w_2, \dots, w_k, \dots, w_K$$

where $\max(m, n) < K < m + n - 1$. The warping path is typically subject to several constraints [33,35]:

1. *Boundary conditions:* $w_1 = (1, 1)$ and $w_k = (m, n)$, simply stated, this requires the warping path to start and finish in diagonally opposite corner cells of the matrix.
2. *Continuity:* Given $w_k = (p, q)$ then $w_{k-1} = (p', q')$ where $p - p' \leq 1$ and $q - q' \leq 1$. This restricts the allowable steps in the warping path to adjacent cells (including diagonally adjacent cells).

3. *Monotonicity:* Given $w_k = (p, q)$ then $w_{k-1} = (p', q')$ where $p - p' \leq 0$ and $q - q' \leq 0$. This forces the points in W to be monotonically spaced in time.

In addition, warping is also subject to global path constraint and slope weighting. There are large numbers of possible warping paths that can satisfy the above constraints, but our interest lies in the path, which is capable of minimizing the warping cost. This path can be found efficiently using dynamic programming to evaluate the following recursion which defines the cumulative cell and the minimum of the cumulative distances of the distance $D(i, j)$ as the distance $d(i, j)$ found in the current adjacent elements:

$$D(i, j) = d(i, j) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\}$$

It is a point worth noting that the possible warping paths grow exponentially with the length of the time series. The distance that is minimized over all paths is the Dynamic Time Warping distance and can be computed using Dynamic Programming in $O(mn)$ time. The details are given in [33,35].

Earlier studies on pattern recognition related problems in speech research have used the DTW algorithm mostly for obtaining the matching score. It ignores the information present in the resulting warping path. The DTW path is represented by a sequence of points, where the frame index of the test cry signal is ai , while the frame index of the reference cry signal is bi . Motivated by the recent study on utilizing optimal warping to estimating duration and pitch information [54], an analysis was carried out to study the nature of the warping path by matching the reference and test cries. It was observed that the nature of the warping path that joins the points follows closely the diagonal line in the plane when cry signal of one normal infant is matched (or warped) with another normal infant, whereas it deviates significantly from the diagonal line for matching with abnormal infant cry. Figure 14.13 illustrates the behavior of the warping paths for normal and abnormal infant cry as test cry (for infant with asthma, HIE, larynx not developed) with a reference cry of normal infant for four cry modes, viz., weak vibration, inhalation, pitch rising, and dysphonation. Since all the cry modes were not present in all three abnormal cases, only four cry modes could be considered for the present study.

Some of the observations from plots are as follows

1. The optimal warping path is *near diagonal* in almost all the four cry modes for the case of normal infant matched with another normal infant. (Fig. 14.13a–d). This means that it requires less warping cost to map cry mode of one normal infant with another. And thus this forms the basis for our method of classifying normal vs. abnormal infant cry.
2. The optimal warping path of weak vibration cry mode for normal infant matched with another normal infant is near diagonal and significant deviation from diagonal (straight line) in warping path is observed for abnormal infants. It seems that this cry could give better clues for infant cry classification. (Fig. 14.13a). This means that weak spectral energy distribution of weak vibration cry mode is very sensitive to any abnormalities in infant cry, again with respect to glottal closure instants and, thus, pitch harmonic structure (Fig. 14.13a–d).

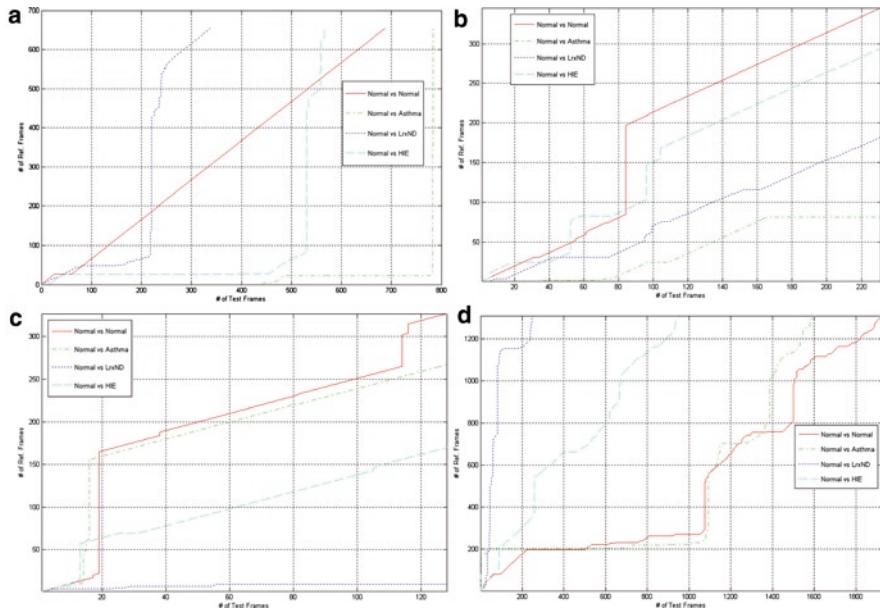


Fig. 14.13 DTW warping path for matched (i.e., normal vs. normal) and mismatched (i.e., normal vs. asthma, normal vs. larynx not developed (LrxND), normal vs. Hypoxic Ischemic Encephalopathy (HIE) for different cry modes: **(a)** weak vibration, **(b)** dysphonation, **(c)** rising, and **(d)** inhalation

3. The optimal warping path *significantly* deviates from diagonal for infant whose larynx is not developed. This indicates strong correlation of warping path deviation from diagonal (straight line) with abnormality in infant cry. This is true in majority of the cases of cry modes considered in Fig. 14.13a–d. This is due to the fact that the DTW algorithm is trying to map (warp) unvoicing feature of infant cry (i.e., larynx abnormalities with respect to glottal closure instants) with voicing feature (pitch rising) and, thus, requires larger warping cost, resulting in larger deviation from baseline diagonal warping path.
4. Abnormalities are evident in particular cry mode and may or may not be evident in another cry mode for the same infant. For example, for infants with asthma, the optimal warping is more aligned along diagonal for the inhalation and pitch rising cry mode, whereas there is significant deviation from the diagonal for the weak vibration and dysphonation cry mode. This means that the several levels of checking are necessary to arrive at a conclusion as to what abnormality could be found in a particular infant cry signal.
5. For infants whose larynx is not developed (i.e., very little or no voicing), the optimal warping path shows maximum deviation from diagonal for pitch rising cry mode. This is again due to the fact that the DTW algorithm is trying to map (warp) unvoicing (larynx abnormalities) with voicing (pitch rising) and hence requires larger cost and results in larger deviation from diagonal. In addition, this suggests that we can exploit the cry mode of pitch rising (i.e., phenomenon

- related to voicing) as a reference template for classifying larynx abnormalities related to glottal closure instants (GCI) (i.e., distance between two consecutive GCIs is the pitch period)
6. Next to abnormalities in larynx, infants with HIE show significant optimal warping path deviation from diagonal in almost all four cry modes.

The significance of this behavior of the optimal warping path for normal and abnormal infant cry can be explained as follows.

When a particular cry mode of a normal infant (test) is matched (time warped) with the same cry mode of another normal infant (reference), then the occurrence of different events rather than sequence of events (in the form of spectral energy distribution) in a cry mode is likely to be *similar* (if not identical) and thus will require less cost for warping and hence this consistency will result in a warping path which is nearly straight, i.e., along the diagonal (as clearly evident from Fig. 14.13a, red colored warping path). On the other hand, when a particular cry mode of an abnormal infant (test) is matched (time warped) with the same cry mode of normal infant (reference), occurrence of different events in cry for abnormal infant is expected to be different than its normal counterpart which in turn will reflect on the relative spectral energy distribution in feature vectors during DTW and thus will require relatively higher cost for warping and hence this inconsistency will result in warping path which is significantly deviated from straight line (as clearly evident from Fig. 14.13a, blue and green colored warping paths).

14.6 Summary and Conclusions

Most of the commercially available products such as baby cry analyzers that classify an infant's level of distress for hungry, bored, annoyed, sleepy or stressed, do not meet the more demanding and much more challenging problem of using infant cry for clinical diagnosis [58]. In this chapter, an attempt is made to explore the potential of spectrographic analysis, a classic method in the area of speaker recognition, for diagnosing and treating neonatal problems and establishing a baseline of normal functioning in the healthy newborn. An analysis of normal and abnormal infant cry is presented using ten distinct cry modes that were observed in spectrograms. It was observed that clinical abnormalities in neonates could be correlated to differences in the spectral energy distribution and the pitch harmonic structure of spectrograms. In addition, an interesting approach is proposed for classification of normal vs. abnormal infant cry by exploiting the nature of the optimal warping path in the DTW algorithm.

The technology addressed in this work is of commensurate social relevance just as it is of diagnostic importance, in that such technological applications may increase confidence in clinical diagnosis of abnormal infant cry, which would likely prompt the physician to take a more proactive role in treatment of a newborn. For example, from spectrographic analysis, following clues can be inferred which may find its use in clinical diagnosis and treatment:

- 1) If the *harmonic structure* is absent or significantly less dominant in spectrogram, then it gives clues for abnormalities related to larynx especially in the context of glottal closure instants, i.e., if the sudden closure of glottis is not there, then there is no impulse-like excitation to the vocal tract and hence it will not reflect a dominant harmonic structure in spectrograms.
- 2) If the spectrograms contain *dominant double harmonic break* or dysphonation, then it can be correlated with muscle pain or discomfort due to some abnormality.
- 3) If the duration of *inhalation* cry mode is greater than the normal prescribed duration for healthy newborns then it is probably normal infant cry whereas small intervals of inhalation followed by voicing indicate breathing difficulty that may correlate to chronic asthma, which is a serious condition that must be carefully watched.

In sum, technology derived from the field of speaker recognition can improve and complement the clinical diagnostic skills of pediatricians and neonatologists, by helping them to detect early warning signs of pathology, developmental lags, and so forth. This is especially helpful today in a healthcare environment where newborns do not have the luxury of being solely attended by one physician, and are, instead, monitored remotely by a centralized computer control system.

Motivated by a need to equalize the level of neonatal healthcare (not every neonate has the luxury of being monitored at a teaching hospital equipped with a high level neonatal intensive care unit), I propose for the next phase of research a quantifiable measurement of the added clinical advantage to the clinician (and ancillary healthcare worker) of a baseline comparison of normal versus abnormal cry. This chapter has served to introduce the reader to a discussion of how spectrographic analysis, developed for speaker recognition, may find another niche in helping clinicians in making better informed diagnostic and treatment decisions when caring for their neonatal patients.

Acknowledgment The author would like to thank DA-IICT authorities for their kind support to carry out this research work. He would also like to thank Ms. Neeharika Buddha, Dr. B. V. Adinarayana (KGH, Visakhapatnam) and Prof. B. Yegnanarayana of IIIT Hyderabad for their kind help and cooperation during this work.

References

1. V. Apgar, “A proposal for a new method of evaluation of the newborn infant,” *Curr. Res. Anesth. Analg.*, vol. 32, pp. 260–267, 1953.
2. J. F. Bosma, H. M. Truby, and J. Lind, “Cry motions of the newborn infant,” *Acta Paediat. Scand. Suppl.*, vol. 163, pp. 61–92, 1965.
3. Neeharika Buddha and Hemant A. Patil, “Corpora for analysis of infant cry,” *Int. Conf. on Speech Databases and Assessments, Oriental COCOSDA*, Hanoi, Vietnam, pp. 43–48, Dec. 4–6, 2007.
4. M. Corwin, and H. Golub, “Medical applications of infant cry analysis,” A. Milunsky, E. Friedman, and L. Gluck, *Advances in prenatal medicine*, New York: Plenum Press, vol. 4, pp. 163–188, 1985.
5. M. J. Corwin, B. M. Lester, C. Sepkoski, Peucker, M. Kayne, H., H. L. Golub, “Newborn acoustic cry characteristics of infants subsequently dying of sudden infant death syndrome,” *Pediatrics*, vol. 96, no. 1, pp. 73–77, July 1995.

6. R. Colton and A. Steinschneider, "The cry characteristics of an infant who died of sudden infant death syndrome," *J. Speech, Hearing Dis.*, vol. 46, pp. 359–363, 1981.
7. K. John Cullen Jr., Nancy Fargo, Richard A. Chase, Peggy Baker, "The development of auditory feedback monitoring: I Delayed auditory feedback studies on infant cry," *J. Speech, Hearing Research*, vol. 11, pp.85–93 1968.
8. G. Fant. *Acoustic Theory of Speech Production*. The Hague: Mouton, 1960.
9. V. Fisichelli, M. Coxe, L. Rosenfeld, A. Haber, J. Davis, and A. Karelitz, "The phonetic content of the cries of normal infants and those with brain damage," *Journal of Psychology*, vol. 64, pp. 119–126, 1966.
10. B. F. Fuller and Y. Horii, "Spectral energy distribution in four types of infant vocalizations," *J. Commun. Disord.*, vol. 21, pp. 251–261, 1988.
11. L. Gray, "Signal detection and analysis of delays in neonates vocalization," *J. Acoust. Soc. Am.*, vol. 82, pp. 1608–1611, 1987.
12. Susan M. Grau, Michael P. Robb, Anthony T. Cacace, "Acoustic correlates of inspiratory phonation during infant cry: Research note," *Journal of Speech and Hearing Research*, vol. 38, pp. 373–381, April 1995.
13. H. L. Golub and M. J. Corwin, "A physioacoustic model of infant cry," in *Infant Crying: Theoretical And Research Perspective*, B. M. Lester and C. F. Z. Boukydis, Eds. New York, Plenum, pp.59–81, 1985.
14. G. E. Gustafson and J. A. Green, "On the importance of fundamental frequency and other acoustic features in cry perception and infant development," *Child Developmerit*, vol. 60, pp. 772–80, 1989.
15. D. Harrison, "Histologic evaluation of the larynx in sudden infant death syndrome," *Annals of Otology, Rhinology, and Laryngology*, vol. 100, pp. 173–175, 1991.
16. O. C. Irwin, "Infant speech: Development of vowel sounds," *J. Speech Hearing Dis.*, vol. 13, pp. 31–34, 1948.
17. F. A. Ismaelli, A. Manfredi, and P. Bruscaglion, "Parametric and non-parametric estimation of speech formants: application to infant cry," *Med. Eng. Phy.*, vol. 18, no.8, pp. 677–691, 1996.
18. C. C. Johnston, and D. O'Shaughnessy, "Acoustical attributes of infant pain cries: Discriminating features," in *Proc. Vth World Congress on Pain*, R. Dubner, G. F. Gebhart, and M. R. Bond (Eds), Elsevier, Amsterdam, 1988.
19. L.G. Kersta, "Voiceprint Identification," *Nature*, vol. 196, pp. 1253–1257, 1962.
20. R. D. Kent and A. D. Murray, "Acoustic features of infant vocalic utterances at 3, 6, and 9 months," *J. Acoust. Soc. Am.*, vol. 72, pp. 353–365, 1982.
21. P. Lieberman, K. S. Harris, and P. Wolff, "Newborn infant cry in relation to nonhuman primate vocalizations," *J. Acoust. Soc. Amer.*, vol. 44, pp. 365 (A), 1968.
22. P. Lieberman, K. S. Harris, P. Wolff and L. H. Russell, "Newborn infant cry in relation to nonhuman primate vocalizations," *J. Speech and Hearing Res.*, vol. 14, pp. 718–727, 1971.
23. J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp.561–580, 1975.
24. K. Michelsson, "Cry analyses of symptomless low-birth-weight neonates and of asphyxiated newborn infants," *Acta Paediat. Scand. Suppl.*, vol. 216, pp. 1–45, 1971.
25. K. Michelsson and P. Sirvio, "Cry analysis in congenital hypothyroidism," *Folia Phoniafica*, vol. 28, pp. 40–7, 1976.
26. K. Michelsson, P. Sirvio, and O. Wasz-Hockert, "Sound spectrographic cry analysis of infants with bacterial meningitis," *Develop. Med. and Child Neuro.*, vol. 19 (b), pp. 309–15, 1977.
27. K. S. R. Murty, B. Yegnanarayana and M. A. Joseph, "Characterization of glottal activity from speech signals," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 469–472, June. 2009.
28. P. E Ostwald and T. Murry, "The communicative and diagnostic significance of infant sounds," chapter 7, *Infant Crying: Theoretical and Research Perspectives*. Plenum Press, New York, NY, pp.139–158, 1985.
29. Hemant A. Patil, "Infant identification from their cry," *7thInt. Conf. Advances in Pattern Recognition ICAPR*, ISI Kolkatta, *IEEE Computer Society*, pp. 107–109, Feb. 4–6, 2009.
30. D. S. Paterson, F. L. Trachtenberg, F. G. Thompson *et al.*, "Multiple serotonergic brainstem abnormalities in sudden infant death syndrome," *J. Amer. Med. Ass.*, vol. 296, no. 17, pp.2124–2132, November 2006.

31. R. Prescott, "Infant cry sound: Developmental features," *J. Acoust. Soc. Amer.*, vol. 57, pp. 1186–1191, 1975.
32. Athanassios Protopapasa and Peter D. Eimas, "Perceptual differences in infant cries revealed by modifications of acoustic features," *J. Acoust. Soc. Am.*, vol.102, no.6, pp. 3723–3734, December 1997.
33. L. Rabiner, A. Rosenberg, and S. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, pp. 575–582, Dec. 1978.
34. Rohilah Sahak, Wahidah Mansor, Lee Yoot Khuan, Azlee Zabidi, Farah Yasmin, "An investigation into infant cry and Apgar score using principle component analysis," *5th Int. Colloquium on Signal Proces. and its Applications (CSPA)*, pp.209–214, 2009.
35. H. Sakoe and S. Chiba, "Dynamic programming optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-26, no.1, pp/43–49, February 1978.
36. R. Stark, and S. Nathanson, "Unusual features of cry in an infant dying suddenly and unexpectedly," J. Bosma and J. Showacre (Eds.), *Development of upper respiratory anatomy and function: implications for the Sudden Infant Death Syndrome*, pp.323–352. Washington, DC: US Printing Office, 1975.
37. L. Sheppard and J. Kaufman, "Sorting out the role of air pollutants in asthma initiation," *Epidemiology*, vol. 11, no.2, pp.100–101, March 2000.
38. Janet L. Tenold, David H. Crowell, Richard H. Jones, Thomas H. Daniel, D. Frank McPherson, Arthur N. Popper, "Cepstral and stationarity analyses of full term and premature infants' cries," *J. Acoust. Soc. Am.*, vol. 56, No. 3, pp. 975–980, September 1974.
39. B. Thach, "The potential role of airway obstruction in sudden infant death syndrome," J. Culbertson, H. Krouse and R. Bendell (Eds.), *Sudden Infant Death Syndrome*. Baltimore: Johns Hopkins University Press, pp. 62–93, 1989.
40. J. Thoden, A.-L. Jarvenpaa and K. Michelsson, "Sound spectrographic cry analysis of pain cry in prematures," chapter 5. *Infant Crying: Theoretical and Research Perspectives*. Plenum Press, New York, New York, 1985.
41. S. Tonkin, "Airway occlusion as a possible cause of SIDS," F. Robinson (Ed.), *Sudden Infant Death Syndrome (SIDS) Canada*: Canadian Foundation for the Study of Sudden Infant Death, pp. 34–97, 1974.
42. H. M. Truby and J. Lind, "Cry sounds of a newborn infant," *Acta Pediatric Scand., Suppl.*, vol. 163, pp.8–59, 1965.
43. Antonio Verdúzco-Mendoza, Emilio Arch-Tirado, Carlos A. Reyes García, Jaime Leybón Ibarra, and Juan Licona Bonilla, "Qualitative and quantitative crying analysis of new born babies delivered under high risk gestation," A. Esposito et al. (Eds.): *Multimodal Signals*, LNAI, Springer-Verlag Berlin Heidelberg, vol. 5398, pp. 320–327, 2009.
44. O. Wasz-Hockert, V. Vuorenkoski, E. Valanne, K. Michelsson, "Sound spectrographic studies of the cry of newborn infants," *Experiencia*, vol. 15, no.18, pp. 583–584, 1962.
45. O. Wasz-Hockert, E. Valanne, V. Vuorenkoski, K. Michelsson, A. Sovijarvi, "Analysis of some types of vocalization in the newborn and in early infancy," *Ann Paediatr Fenn.*, vol.9, pp.1–10, 1963.
46. O. Wasz-Hockert, T. Partanen, V. Vuorenkoski, E. Valanne, and K. Michelsson, "The identification of some specific meanings in infant vocalization," *Experiencia*, vol. 20, pp. 154–156, 1964.
47. O. Wasz-Hockert, J. Lind, V. Vuorenkoski, T. Partanen, and E. Valanne, "The infant cry- A spectrographic and auditory analysis", *Spastics International Medical Publications in Association with William Heinemann Medical Books Ltd.*, 1968.
48. O Wasz-Hockert, K. Michelsson, and J. Lind, "Twenty-Five Years of Scandinavian Cry Research," chapter 4. *Infant Crying: Theoretical and Research Perspectives*. Plenum Press, New York, New York, 1985.
49. Wikipedia: Sudden Infant Death Syndrome.
50. P. H. Wolff, "The natural history of crying and other vocalizations in early infancy," *Determinants of Infant Behavior IV*, M. Foss (Eds), Methuen, London, U.K. 1969.

51. Q. Xie, R. K. Ward and C. A. Laszlo, "Automatic assessment of infant's levels-of-distress from the cry signals," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 4, pp. 253–265, July 1996.
52. Q. Xie, R. K. Ward, and C. A. Laszlo, "Determining normal infants' level-of-distress from cry sounds," *Proc. of the 1993 Canadian Conf. on Electrical and Computer Engineering*, pp. 1094–1096, Vancouver, B. C., September 14–17, 1993.
53. Naoto Yamane and Yoko Shimura, "The acoustic characteristics of infant cries," *J. Acoust. Soc. Am.*, vol. 120, no. 5, pt. 2, pp. 3136, November 2006.
54. B. Yegnanarayana, S. R. M. Prasanna, J.M. Zachariah and C. S. Gupta, "Combining evidence from source, suprasegmental and spectral features for a fixed-text speaker verification system", *IEEE Trans. Speech Audio Processing*, vol.13, no.4, pp.575–582, July 2005.
55. P. Zeskind, and B. Lester, "Acoustic features and auditory perceptions of the cries of newborns with prenatal and perinatal complications," *Child Development*, vol. 49, pp. 580–589, 1978.
56. P. Zeskind and B. Lester, "Analysis of cry features in newborns with differential fetal growth," *Child Development*, vol. 52, pp. 207–212, 1981.
57. T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practices*. Pearson Education, 2002.
58. Why Cry @ Baby Crying Analyzer: <http://www.showeryourbaby.com/whycrbacran1.html>.

Epilog

James A. Larson

Predicting the future is risky. Many predictions never come true, and really interesting things happen that were never predicted. Having a target idea is useful to aim for. So, having looked over this book detailing some of the most significant advances in speech recognition today, the following is my best vision of what the future will bring.

Four Major Trends

New technology changes the way people interact with computers. PCs enabled people to use a keyboard and screen rather than review printed reports. The graphical user interfaces introduced by the Xerox Star, Apple Macintosh, and Microsoft Windows brought computing power to millions of users. Portable computing now enables users to access information and to interact with others practically anytime and anywhere. It seems the rate of change is accelerating. Customer acceptance of mobile devices has been phenomenal.

The biggest use of speech technologies will be in multimodal applications – applications in which users cannot only speak and listen, but also gesture and see. The black phones that we grew up with are being replaced by mobile devices that can be used for much more than just speaking and listening. These devices will constitute the world where speech technologies are used.

For this reason, I foresee four major trends that will shape the future of speech-enabled computer–user interaction:

New Form Factors

A mobile phone is like a Swiss Army Knife: both have multiple, sometimes unrelated, uses. Having all these functions in a single device is convenient. However, the mobile device may be awkward to use. Users constantly reposition it between

their ear and their eyes. The small screen is overloaded with far too many menus and options. I predict that the current mobile device will explode into multiple components that can be connected together in various ways, much like different the components of Lego, tinker toy, or erector sets that can be combined to configure a large variety of interesting configurations. The mobile phone will be replaced by a personal server, microphones, speakers, cameras, and displays that are physically separate from each other, but are wirelessly connected.

Personal Server. The personal server will be about the size of a deck of cards and can be carried in a shirt pocket or purse. It will contain computer memory and will be able to send and receive information with other components. This personal server will store many kinds of data and information, possibly including the following:

- A backup of your files from your PC;
- Pictures and video from your digital camera;
- Music downloads from the Internet;
- Frequently visited Websites on the Internet;
- Personal information, including medical, identification, and contact information for relatives and friends, as well as personal account information;
- Software agents, and their actions and results.

A personal server connects to the Internet, and accesses a nearly endless collection of data and information.

Microphones and speakers. Speech will be the primary means for interacting with software agents residing in the personal server. There will be a variety of microphones and speakers embedded into personalized jewelry, so users can speak and listen to software clients without repositioning their head or hands. Earrings, eyeglass frames, and hearing aids will contain speakers. Necklaces, lapel pins, and glass frames will contain microphones. Some microphones may also include a small camera that captures lip movements, which improves speech recognition accuracy.

Cameras and displays. Information can be displayed to the user using a variety of components:

- A small, portable display attached to a wristband (reminiscent of the illustrations of cartoonist Chester Gould who introduced to the “Dick Tracy” comic strip the two-way wrist radio in 1946 and the two-way wrist TV in 1964) or worn on a chain around the user’s neck.
- Existing computer and television screens that use wireless communication to transfer information to any local display for presentation to the user, and as the user moves from room to room, the presentation continues but on different screens in different rooms.
- A micro projector, possibly attached to a key chain that projects images onto a convenient surface, such as a blank sheet of paper or even a paper form with no data.
- Eyeglasses that contain a display. Superimposed on the lens of the eyeglass so that the user can see both the display and the real world at the same time.

Other devices may include a General Position Sensing (GPS) device that detects the location of a user; orientation devices that determine the direction the user is

looking; and other biometric instrumentation that determine what the user is feeling. Most notable, attaching all of these devices will become as simple and natural as dressing in the morning.

These components allow computer users to escape from the “office position” – sitting in front of a computer with their hands on the keyboard – so that they can move from place to place in the office building, in the city, or in the world. No longer will users need to go to the computer; instead, the computer will always be with the user – just as a wallet or watch is always with its owner.

Connectivity

Cell phones and personal digital assistants (PDAs) enable users to connect with other users as well as information on the Internet from whatever locations and whenever necessary. In the future, connectivity will be like electricity: it is “just there” and is sorely missed on the rare occasions when not available. Given the ubiquity of connectivity, users will be able to switch seamlessly among communication networks as they move from place to place in their daily lives.

Connectivity will enable users to interact with data on their personal server, with personal data stored on servers anywhere on the Internet, and with Web pages on the World Wide Web. Software agents will manage data: placing frequently used data in the personal server, storing backup and less frequently accessed personal data to be safely stored on a trusted remote server, and accessing general information from the World Wide Web. Software agents will apply voice search, filtering, and prioritizing algorithms to manage these data on behalf of the user.

And, of course, connectivity will enable any user to interact with others, either by voice, text, video, or other modes of communication.

Multimodal

For many years, radio was the main form of information and entertainment in the home, enabling users to listen to people and events from outside their homes. Telephones enabled users to both speak and listen, and television enabled users to both listen and see. Modern technology now encourages users to speak, listen, see, and use other modes of interaction. In addition to using more of the user’s senses, multimodal technology provides a faster bandwidth for information exchange.

Alternative input modes such as speaking, pressing keys, touching a screen, or possibly glancing at an object may act as backup modes for one another. For example, if speech recognition fails in a noisy environment, the user can press keyboard buttons. Users may also select the appropriate input mode for their current situation: e.g., speech recognition while walking, or handwriting or key input during a business meeting.

Software Agents

New classes of multimodal applications, called software agents, are emerging. Software agents can be classified as follows:

Active listening. While radios and TVs enable users to listen passively, software agents will enable “active listening.” This is so because users speak commands to start, stop, fast forward and rewind, select content, and increase and decrease speed, volume, and other characteristics of content presentation. As a result, users will be able to navigate the content using voice menus, pick lists, and voice-invoked hyperlinks. In sum, remote controls will be replaced by microphones.

Command and control agents. These agents respond to application-specific commands beyond the active listening commands. A software assistant (or agent) listens for and acts upon user requests. Examples of command and control agents include the following:

- A violin tuner that presents the audio tone after a user says the name of the note, leaving both hands free to tune the violin.
- A TV controller that changes channels, volume, and TV display characteristics.
- An environmental controller that adjusts the temperature, lighting, and security system.
- A family-activity coordinator that enables family members to coordinate their individual activities.

Synthetic agents. Users may converse with software agents representing real or imaginary people, much like an interview or question/answer session. For example, users could ask an artificial Albert Einstein about his life and work; a synthetic pop star about her latest song; or a fictitious 2020 Secretary General of the United Nations about major world events. Speech-enabled video games allow users to speak with other human and artificial players to affect the outcome of games. This technology will also be used for training and educational purposes.

Remember the *Choose Your Own Adventure* books in which the reader could select alternative pages and then read alternative scenarios. The same idea has been applied to the movie film “Last Call by Thirteenth Street.” Audience members are called on their cell phones and encouraged to speak instructions to the on-screen actors. The scenes are automatically switched based on input from the audience members. Multi-user dramatic performances experience this type of interactive participation, which could become available to other forms of entertainment. Imagine a radio station that calls you and asks you what song you want to hear, and then plays that song on the radio.

Develop your own content. Developing and sharing content is a growing activity on the Internet. In addition to passively observing Internet content, users actively add to the content by uploading their pictures to flickr.com, share their thoughts in blogs and wikis, or upload their videos to YouTube.com. Readers rate books on Amazon.com, and post real and fantasy personas on MySpace.com and Facebook.com. Twitter enables fast and wide communication among many users.

Many users contribute content to the same Web page over a period of time. An example is the widely used wikipedia.com, a handy online reference replacing the traditional bulky and expensive encyclopedias. Hundreds of volunteers interactively create, review, and update the expanding encyclopedic content.

Interactive voice dialogs have created software agents that speak and perform actions, rather than listen passively. Content authors become more like playwrights and less like reporters, while users become more like actors than observers. With interactive content, the boundaries between audiences and creators blur, as lectures become conversations, reports are morphed into discussions, and stories transform into activities.

Create your own software agent. As Lev Grossman noted in his *Time* feature article, “Time’s Person of the Year: You” (December 13, 2006), the World Wide Web has become “a tool for bringing together the small contributions of millions of people and making them matter.” Users will continue this trend by creating applications involving multimedia and multimodality. Students will no longer write essays and papers, but instead create software agents that explore alternative viewpoints, such as opposing opinions of the westward movement in America as expressed by avatars representing a frontiersman, a Native American, and a settler.

Future Opportunities

Many new types of applications will be possible using speech technologies for mobile devices, call centers, and clinics.

Mobile devices. Mobile devices will contain software agents that assist you and make your life easier. Many new voice-oriented software agents will answer your questions, including the following:

- Where is it? – When your keys are missing, ask aloud “where are my keys?” The lost key chain blinks or buzzes.
- List reminder – When you need toothpaste, just speak: “Add toothpaste to my shopping list.” The next time you are in a store that sells toothpaste, your mobile device not only reminds you to buy toothpaste, but tells you where to find the toothpaste in the store.
- Who is he? – When you speak to someone you recognize but cannot think of the person’s name, the personal server uses a speaker recognition system to identify the mystery person and speech synthesis to whisper the mystery person’s name in your ear.
- Tell me about it – When you wish to obtain additional information about a landmark such as a statue of Sacagawea encountered while traveling, just ask, “Tell me more about Sacagawea.” Your personal server whispers a short paragraph about the life of the Shoshone woman who guided the Lewis and Clark expedition.

Because people vary the speed, volume, and pitch of their speech, speech is more expressive than text. Speech is also faster and more convenient than typing. We use our voices everyday to provide content when we interact with others.

Speech recognition converts speech to text, and speech synthesis converts text to speech. Speech and text will be interchangeable. Just imagine how voice content will enhance the Web:

- Goods and services ratings – Web site visitors could speak comments and critiques for the benefit of other users. A speaker's tone conveys opinions and feelings about a product or service. The resulting Web site experience would be similar to shopping with several friends.
- Audio annotation – Web sites could offer spoken commentary or individualized audio tours of a Web site and suggest alternatives or advice for the Web site's content.
- Commentary – Web site visitors could express their views in townhall discussions or contribute short stories and anecdotes to a comedy Web site and become an online version of Jay Leno or David Letterman. A voice-based wiki would be more interesting than a text-based wiki because of the emotion expressed in contributors' voices.
- Traffic and news reports – People could call to report traffic conditions at various locations using their hands-free mobile device. Radio and Internet listeners would hear the most recent messages and adjust their routes accordingly. Anyone can become a reporter by phoning in eyewitness accounts of emerging news events along with pictures captured by their cell phones.
- Celebrations – User groups capture and collect audio and video from members of a family or group about topics, events, or holidays. These audio memories can be fondly reviewed years later.
- National landmarks, museum exhibits, and other frequently visited monuments – Web site visitors could inform others where to see bears in Yellowstone Park, to locate the secret symbols at a tourist site referenced in a popular book or movie, and reminisce about places visitors played in a grassy meadow that is now a parking lot.

Users repeatedly return to Web sites after contributing content because they want to learn how others respond to the users' contributions. Generally, the aggregate of several individual contributions is more informative and useful than individual contributions, which is why wikis are so popular today. Voice content contributed by Internet users will have a similar effect, making Web sites more interesting and compelling.

We have already seen an explosion of person-to-person voice communications on the Internet due to VoIP technologies. The future will bring a dramatic increase in voice interaction with Web content. That is, while we only listen to radio and listen and see TV content, we will speak and listen – interact – with Internet content.

Call centers. Telephone operators and receptionists are now mostly automated. So it seems reasonable that call centers can be automated, although I am not sure that they should be automated entirely.

Printed documentation has largely been replaced by online help. However, the online help often is not very helpful. New forms of voice-enabled automated help will assist users in performing a variety of vexing tasks:

- Assembly – Every product will have a Web site where users can access online software agents, especially for products labeled with “some assembly required.” It would be nice to have an automated help software agent to direct you with step-by-step instructions for assembling your child’s holiday gift the night before the holiday begins. Because your hands are busy, you can verbally request that each instruction be read to you, perhaps augmented with an illustration. Online software agents could handle most of the routine trouble calls, leaving human experts to handle the challenging problems.
- Debugging and troubleshooting – The automated help software agent asks you a series of questions to identify what the problem is, and then provides step-by-step instructions for repairing the problem.
- Scheduling and rescheduling – Arrange the time and place for appointments and services. Many airlines now automatically call passengers to inform them of flight delays. Doctors’ offices and home repair/installation companies should also call customers to notify them of delays or to reschedule service calls. Although some utility companies or major appliance companies have started to use automated voice messages to inform clients of service call delays or cancellations, in the future automated help software agents will, on a routine basis, perform a wide array of scheduling tasks.
- Order entry and status – Customers will be able to shop online, view products, talk with automated sales agents, talk with co-shoppers, and review comments from customers who have previously purchased the product or service. Customers can check the current status of repair jobs, the progress of custom construction jobs, and delivery status, among other things, with the assistance of the voice-enabled automated help software agent.
- Account questions – Customers will be able to ask for not only their current balances, but also credit and debit charges, extra fees and account discrepancies, or any other major issue related to their account.
- Strengthen customer loyalty – In addition to helping customers solve their current problem, call centers also act as a sounding board for customer complaints, strengthen customer loyalty, and make customers feel part of the company’s larger community. It is not clear to me how much of this can be automated; but years ago, it was also not clear to me how stenographers could be replaced by dictation software.

Clinics. In addition to the call center functions of scheduling appointments, ordering medicine, and resolving account questions, two additional services will improve patient care by providing better service, at faster rates and at lower costs:

- Remote diagnosis – Many patients are not able to travel to the doctor, including home-bound patients who do not drive, live in remote areas, or live in areas with limited health care, especially specialty health care. Wireless services which can be connected to the patient’s mobile device can capture the patient’s temperature, blood pressure, glucose level, and other vital signs. Patients can answer questions about their medical histories, and their current health concerns including “where it hurts.” Medical personnel can use video to view wounds and

observe the patient's general condition. While I personally feel that a software agent may diagnose what ails me correctly, I would like a human specialist to confirm that diagnosis. Nevertheless, the software agent steps in and performs vital preliminary tasks for the overworked physician who cannot possibly attend to the patient at that very moment.

- Remote monitoring – Rather than the patient traveling to the clinic or medical personnel traveling to the patient, the patient's mobile device can periodically monitor the patient's progress, ask the patient how he or she feels, and report this information to the patient's medical clinic. If the software agent detects a possible problem, the patient is connected with a healthcare professional for further discussion. And most importantly, heart attacks, strokes, and other emergencies can be detected and emergency personnel dispatched.

The goal is not to replace trained medical personnel, but to enable them to perform their jobs more efficiently by offloading routine tasks to automated agents. Patients receive care when they need it, often without time-consuming and expensive travel.

Our Responsibilities

As experts in speech technology, we have responsibilities to provide usable and safe products and services for anyone who uses our products.

User interface best practices and guidelines. When fonts were first introduced, many messages looked like ransom notes from kidnappers. When color was introduced, many reports looked like they barely survived an explosion in a paint factory. To avoid these annoying user interfaces, developers adopted suggestions and best practices for using fonts and colors.

With the introduction of multiple modes of input – voice, pen, and keys – inexperienced developers may design loud, confusing, and annoying user interfaces that would result in low-user performance and high-user discontent. While many suggestions for guidelines for speech-only user interfaces exist, there are fewer suggestions for guidelines for multimodal applications that include speech. A first attempt to define multimodal guidelines is “Common Sense Suggestions for Developing Multimodal User Interfaces” [<http://www.w3.org/TR/2006/NOTE-mmisuggestions-20060911/>]. As we gain more experience with multimodal inputs, the industry needs to adopt user interface guidelines so that users can transfer knowledge and skills learned from one user interface to other user interfaces.

Usage safeguards. Many people fear technology. The constant monitoring of people in the book *Brave New World* and the computer HAL's take over of the space explorer's mission in the film *2001: A Space Odyssey* left me with a certain distrust of automation.

Speech technologies, as with most technologies, can be used for both good and bad. While speech technologies enable greater connectivity among users and with

information sources on the Internet, people who use new technologies for fraudulent purposes may in fact take advantage of new technologies to impose their will on others. Sharing some information is generally good, but sharing too much information can lead to theft and other serious problems.

It is our responsibility as technology developers to encourage the appropriate use of our technology.

This collection of essays and research papers edited by Amy Neustein illuminates many significant advances in speech recognition in mobile environments, call centers, and clinics. There is no question but that these advances will continue into the future. Clearly, we must be responsible for establishing the political, legal, and technological safeguards that avoid the catastrophes that can result from our inventions; otherwise, advances in speech recognition, no matter how promising, will undermine the very purpose they were designed to serve.

James A. Larson, Ph.D., is a speech application consultant for Larson Technical Services, VoiceXML trainer, and co-chair of the World Wide Web Consortium's Voice Browser Working group, which develops language standards for speech applications. He is also the co-program chair of the annual SpeechTEK conference. Dr. Larson is the author of many frequently cited technical papers on user interfaces; he teaches courses in building speech applications at Portland State University and Oregon Institute of Technology.

About the Author

Amy Neustein is Editor-in-Chief of the *International Journal of Speech Technology* (Springer Verlag). She is Founder and CEO of *Linguistic Technology Systems*, a NJ-based think tank for intelligent design of advanced Natural Language Understanding software to improve human response in monitoring recorded conversations of terror suspects and of customers' calls into contact centers. Neustein is a graduate of Boston University where she received her Ph.D. in sociology; her specialty area is Conversation Analysis. She has published a number of scholarly articles, chapters, and books, and is the recipient of a Pro Humanitate Literary Award. She serves as a moderator and panelist at academic and industry conferences, and is a member of MIR (Machine Intelligence Research) Labs.

Index

A

acoustic audio descriptors
 formants, 196
 harmonics to noise ratio (HNR), 196
 intensity, 196
 loudness, 196
 Mel-Frequency Cepstral Coefficients (MFCCs), 196
 pitch, 196
 Support Vector Machines (SVM), 196
acoustic classifiers, 205, 210, 218
acoustic misalignments, 68
acoustic model
 grow from simple models, 69 (*see also* single-Gaussian context-independent systems)
 spectral characteristics of, 69
 training of, 69–71
acoustic modeling, 66, 69–71, 254. *See also* language modeling
acoustic-phonetic information, integration into, 277, 279
acoustic streams, 93, 104
active coping, 308, 309
agent
 performance, 135, 137, 141, 143, 236, 237, 239, 240, 242
 productivity, 116, 123
 satisfaction, 124, 136, 138–139, 144
agent-based multimodal interface, 149
agent-enabled transaction, 131
AIMLBot, 100
Amazon’s Kindle, 26
American Medical Transcription Association, 249
Android phone, 58, 80
application server, 159, 161, 164
Armed Forces Longitudinal Technology Application (AHLTA), 254
Artificial Intelligence Markup Language (AIML), 93. *See also* AIMLBot

Asperger syndrome, 312. *See also* Pervasive Developmental Disorder Not Otherwise Specified (PDD-NOS)
ASR
 accuracy, 137, 143, 151 (*see also* re-prompts, number of)
 errors, 132, 136, 145, 166, 213
 misrecognitions from, 207
asthma
 infants with, 333, 342, 343
 spectrographic analysis of infant cry, 333, 336, 337
audio search, 13, 222–228, 230, 231, 235
audio streams, 240
AURORA project, 93
AURORA training set, 104. *See also* TIDigits
authentication path problem, 123
Autism Diagnostic Observation Schedule (ADOS), 313
Autism spectrum disorder (ASD), 305–307, 312–315, 317, 318
Automated directory assistance, 17
Automatic learning agent, 102–104
Automatic learning module, 103
automatic speech recognition and single channel speech separation, lattice rescoring for, 277
automation rate (deflection rate, completion rate), 163–164
Average Handling Time (AHT), 122, 124, 133–134, 136–144, 146, 148, 150, 152, 164, 168, 177, 238–240, 242
average query length, 87

B

baby cry analyzers, 344
back-end SR, 255, 257, 263, 267
background speech recognition, 256, 273

Baker, James and Janet, 251. *See also* Dragon Systems
 barge-in, 200, 208–211, 218
 Berlin Emo-DB (Emotional Speech Corpus), 312
 beta testing
 high-touch, 54
 low-touch, 54–55
 blind and visually impaired, users who are, 23
 botmasters, 100
 ALICE 2005, 110 (*see also* botmasters)
 chat bots, intelligent, 100
 breast cancer survivors, prosodic cues for the assessment of coping, 307
 Brooke, John (Digital Equipment Corporation), 55

C

call center supervisor, escalate call to, 156
 Caller Archetypes, 183, 189
 caller expectation, exceed, 187
 caller experience index, 159, 177–178
 caller frustration, 117
 callers' goals and expectations, 156, 169, 170
 call logs, 158
 Carnegie Mellon University (CMU), 101, 223.
 See also PocketSphinx engine
 Carolinas Medical Center-NorthEast (Concord, NC), 259
 cepstral coefficients, 105, 110. *See also* Mel-Frequency Cepstral Coefficients (MFCCs)
 cepstrum, 94, 95
 cepstrum analysis, 325, 326
 channel transmission errors, 93
 cloud-based computing, 61. *See also* cloud, delivery from
 cloud, delivery from, 62
 cognition, 316, 319
 cognitive load, 109
 command-based dialogue, 110, 194. *See also* fixed commands
 computationally-intensive subsystem, 200
 computer-aided medical transcription (CAMT), 258, 260
 computerized physician order entry (CPOE), 264, 270
 constrained grammars, 49
 contact center, 16, 155–178, 222, 223, 227, 232, 233, 235–243
 conventional DSR front-end that uses only MFCCs, 106. *See also* multi-stream approach
 conventional MFCCs, 93

Convergys call centers, 116–117, 152
 Convergys Human Factors Lab, 130, 133
 cross-channel analytics, 222
 Cry mode, 326, 331–345
 CSAT indicators, preference score of, 151
 customer care, 124, 155–178, 191–219
 Customer relationship management (CRM), 27, 117
 customer satisfaction, 124, 134, 139, 144, 236–238, 240, 241, 292
 customer satisfaction ratings, 239
 customer service, quality of, 157
 Customer Service Representatives (CSRs)
 caller preference for, 182
 expense of, 182
 need to transfer to, 182

D

data connectivity, 64
 Datamonitor, 4
 depression, 305, 307, 308, 310–311, 319
 DES database (Dutch speech), 196
 diagnostics, speech recognition (SR) technology as a tool for, 261
 dialectal variations, 62, 242
 dialog (dialogue) manager, 157, 159, 193, 194, 208
 dictation task, 15
 digital recording device, 255
 direct dictation, 261, 267
 directed-dialog menus, 185. *See also* Semantic Language Models (SLMs)
 directory assistance (411 in the U.S.), 63
 disabilities, users with, 100
 disabled
 community of users, 23
 users who are, 33 (*see also* blind and visually impaired, users who are)
 discrete cosine transform (DCT), 202
 Distributed Speech Recognition (DSR), performance of, 93
 double harmonic break, 326, 331, 332, 335, 337, 339, 345
 Dragon Dictation, 25. *See also* Dragon Naturally Speaking
 Dragon Naturally Speaking, 25, 301. *See also* Dragon Systems
 Dragon Systems, 251. *See also* Baker, James and Janet
 DTW algorithm, optimal warping path of, 326, 341, 343, 344
 dual tone multiple frequency (DTMF), 118, 192, 208, 235

- dynamic decisioning rules, 149
dynamic programming (DP) algorithm, 341
dynamic time warping (DTW) algorithm, 279, 326, 341–344
dysphonation, 326, 331, 332, 337, 339, 342, 343, 345
- E**
elastic models, 71. *See also* Gaussian mixtures per state
electronic documentation
speech-assisted transcription, 267 (*see also* back-end SR)
speech-driven (speech-enabled) EMR systems, 267 (*see also* front-end SR)
electronic health records (EHR), 248, 271, 272
electronic information systems
CPOE, 270
PACS, 270
RIS, 270
electronic medical records (EMR)
adoption rate, 264–266
benefits of, 264
speech enabled, 267
electronic patient narratives, 266–267
emotion
class, 205, 206, 215, 218
recognition, 196, 205, 311
speech transcripts, annotating in, 319
emotional salience, 205, 206, 218
emotional state, computational modeling of, 306
emotion related state, 306. *See also* speaker state
EMR Adoption Model (EMRAM), 265. *See also* Health Information Management Systems Society (HIMSS)
endpointer, 80
endpointing, 22, 80
end-user, 12, 62, 67, 116, 122, 149, 151, 152, 235, 236, 275–301
environmental health officers, 280
Epidemiological Wizard, 282
Erlangen–Nürnberg, 316
error handling, 125, 128, 135, 139, 146, 148, 151
ETSI standard DSR-XAFE, 105
European Telecommunications Standard Institute (ETSI), 93. *See also* AURORA project
eXtended Audio Front-End (XAFE) as coder-decoder (codec), 93. *See also* ETSI standard DSR-XAFE; 3G Partnership Project (3GPP)
- F**
Fair Debt Collection Practices Act (FDCPA), 241
Fant's acoustic theory of speech production, 331
first contact resolution (FCR), 237, 238
fixed commands, 100, 109
formant
formant contour, 325
formant frequencies, 94–96, 279, 324, 325, 330, 331 (*see also* spectrograms)
formant-like (FL) features, 93–96, 105
front-end
process, 93
speech recognition, 93, 253, 255–257, 261, 263, 267
- G**
garbage turns, 213
utterances, 198
Gartner, Inc, 6
Gaussian mixtures per state, 105
Genetic Algorithms (GAs), 98, 100, 104, 106–108, 110
genetic operators, 98–100
Global system for mobile (GSM), 104, 225, 230, 233
glottal closure, 342–345
glottal closure instants (GCI), 342–345
Goals, Operators, Methods and Selection (GOMS) Model, 125
GOOG-411, 63–64, 66, 69, 70, 76, 77. *See also* triphone systems grown from decision trees and use GMMs with variable numbers of Gaussians per acoustic state
GOOG-411 (800-GOOG-411), 63
Google
Android open-source mobile phone operating system, 7
Open Handset Alliance, 7
Google Maps for Mobile (GMM), 64–65, 70
Google Mobile App (GMA) for iPhone, 65, 79–82
Google Search by Voice (search by voice), 58, 61–89. *See also* Microsoft's voice-enabled Bing
Gottschalk-Gleser scales, 308, 309, 317
3G Partnership Project (3GPP), 93. *See also* *eXtended Audio Front-End* (XAFE) as coder-decoder (codec)
GPS, 10, 28, 29, 120, 182
capability, 10
graphical display, 77, 330

graphical user interface (GUI)

- overburdened, 5
- workstation, overlaying multimodal onto, 125

GUI-based transaction, 131

H

hands-free troubleshooting, 280

hang up, callers who, 163, 165–166, 168, 177, 192

haptic (“touch-based”), 28, 43

Healthcare

- background speech recognition in, 256–257
- front-end speech recognition (real-time speech recognition) in, 255–256
- mobile applications for, 271
- need to document procedures, 4
- real-time SR in, 252, 255–256

Health Information Management Systems Society (HIMSS), 265

Health Information Technology for Economic and Clinical Health Act (HITECH), as part of American Recovery and Reinvestment Act (ARRA), 265

Health Insurance Portability and Accountability Act (HIPAA), 241, 319

Health Level 7 (HL7) data interfaces, 262. *See also* radiology report

Help Requests, 150, 151, 210

hidden agent approach, 149

hidden facts, treated as observable, 158

hidden Markov model (HMM)

- multi-stream, 98, 105
- single state, 105
- whole-word, 105

Hierarchical Language Models (HLM), 49, 50

high-touch and low-touch, 54. *See also* beta testing

homeostasis, strive to maintain, 19

hospital-based medical transcriptionists, 249. *See also* offshore transcription industry

human

- annotator, 161
- transcriber, 12, 69, 161

human computer interaction (HCI), 109, 119–120

human factors issues, 116

human transcription, 68

hyperphonation, 326, 331, 337, 340

I

IGR ranking, 203, 208, 210, 211, 217

impaired, users who are, 23, 34

independent duty corpsmen (IDCs), 280, 294

index speed, 230, 232, 233

industrial hygienists, 280

Infonetics Research, 6

information gain ratio (IGR), 203, 210

In-Grammar, 37, 166–168. *See also*

Out-of-Grammar

inhalaition

pattern, 323

small intervals of, 345

input speech, compressed representation

of the phonetic content of, 225

Institutional Review Board (IRB) process, 276

Interactive Voice Response (IVR), 43, 44,

- 63–65, 92, 123, 124, 140, 141, 150, 157, 159–161, 163–165, 168, 170, 171, 177, 185, 186, 191, 192, 195, 197, 205, 208, 212, 217, 218, 306, 311

platforms, automated call centers that support, 156

system, designers of, 185

inter-rater agreement, 199

iPhone, Blackberry, Android, Symbian, Windows Mobile Devices, 25

J

Joint Commission on Accreditation of Healthcare Organizations, 249

Joint Military Medical Command of the US Department of Defense, 284

K

key performance indicators (KPIs), 241

Keyword (word) spotting, 205, 224, 228, 232

KLAS report, 254, 256, 269

Kurzweil Clinical Reporter, 252. *See also* Kurzweil Computer Products, Inc.

Kurzweil Computer Products, Inc., 251. *See also* Kurzweil, Raymond

Kurzweil, Raymond, 251

L

Language ID, 233, 234

language modeling, 36, 49, 64, 66, 68–76, 166, 223, 229, 242, 252, 257, 258, 261, 311

language model, poor predictions of, 68

Large scale language models, 73–76

- large vocabulary continuous speech
recognition (LVCSR), 223, 224, 226,
228–230, 279
- Laryngomalacia, 324, 326, 337–339
- Latent Semantic Analysis (LSA), 311
- LDC (Linguistic Data Consortium) Emotional
Speech Corpus, 312
- learning
supervised, 69, 197
unsupervised, 63, 69
- Likert scale, 55, 138, 145, 283, 285, 297, 298
- linear discriminative classification (LDC), 197
- linear prediction (LP), 96, 325, 327–329
- Linear Predictive Coding (LPC) analysis, 94,
96. *See also* Line Spectral Frequencies
(LSFs), the set of
- Line Spectral Frequencies (LSFs), the set of,
94. *See also* Linear Predictive Coding
(LPC) analysis
- Linguistic Data Consortium (LDC), 312. *See
also* LDC (Linguistic Data Consortium)
Emotional Speech Corpus
- Linguistic Inquiry and Word Count (LIWC)
paradigm, 309
- live agent, 118, 158, 177
- live and recorded calls, mining of, 242
- live calls, 120, 132–135, 139–140, 147
- live delivery service calls, 136
- live help, wait time for, 187
- live operator, 63
- localization, 10
- logarithmic frame energy (log-energy), 105
- log entries, 159, 161
- logistic regression, 307
- low bit rate speech coding, 93. *See also*
channel transmission errors
- low-vision, 33
- LPC filter, 105. *See also* LPC filter
coefficients LSF vector
- LPC filter coefficients LSF vector, 105
- LP-derived spectral features, 329
- LP spectrum, 327–329
- M**
- machine learning algorithm, supervised
artificial neural network (ANN), 197
- Nearest Neighbor, 197
- Rule Learning, 197
- Support Vector Machine, 197
- mapping language cues to medical
conditions, 306
- Measuring the User Experience* (Tom Tullis
and Bill Albert), 55
- medical dictation, 12, 249, 269
- medical encounter, duration of, 287, 288, 293
- medical transcription, 248–250, 252, 256,
258–261, 267, 269, 270. *See also*
medical dictation
- medical transcriptionist
role of, 248
- staffs, productivity of, 248
- Mel-Frequency Cepstral Coefficients
(MFCCs), 92–95, 98, 105–107, 196,
202, 203
- Mel-Frequency Spectral Coefficients
(MFSCs), 311
- menu driven, 119
- mesh-up databases*, 161
- metric
single, 159, 168
summary, 168 (*see also* True Total (*tt*) and
True Confirm Total (*tct*))
- Microsoft's voice-enabled Bing, 58
- military medical environment, 284
- misrecognition, 40, 43, 44, 67, 80, 140, 195,
200, 201, 207, 209, 211, 287, 300
- misrecognitions at word level, 67
- MIT Media Lab, 315
- mixed-initiative approach, 119
- Mobile
applet, 27
computing, 89, 282
ecosystem, 19–30 (*see also* homeostasis,
strive to maintain)
- mobile Internet, 8–10, 13
- search product, 46
- speech interface, ubiquitous deployment
of, 59
- user interface, last remaining barrier to
applications and services, 31
- voice search, 62, 77
- Mobile Internet Report* (Morgan Stanley), 8
- modality
comparisons, 129
- input, 64, 89
- output, 64, 65
- Modality Thrashing, 151
- Modeling
intent, 51
- language, 66, 72–76, 252, 257, 258, 261
- speaker state, using computational
approach for, 305–319
- statistical, 51, 257
- modern signal processing techniques, 325
- Moss Rehab Hospital (Philadelphia, PA), 315
- MossTalkWords, 315, 316. *See also* Moss
Rehab Hospital (Philadelphia, PA)

- MP3 music players, 27
- Multimodal (multi-modal)
- dialog model, 127
 - feedback, 40, 43
 - input, 47
 - interface, 31–59, 63, 65, 79–81, 116, 117, 120, 123–125, 127, 131, 133–142, 145, 146, 149
 - platforms, 63, 144 (*see also* smartphone)
 - service, 29, 120, 122
 - successive refinement of, 115
 - user experience, 116, 150
 - user interface, 64, 76–78
 - voice search, 65
 - workstation for call center agent, 115, 151
- multimodal interface, business value of
- Accretive Value, 124
 - AHT Savings, 124
 - Earnings Per Share (EPS), 124
 - Net Present Value (NPV), 124
 - payback time, 124
 - Time to Market (TTM), 124
- multimodal performance, 143
- multimodal UI streamlines, 127, 135
- multimodal (multi-modal) user interface (MMUI), multimodal UI, 64, 76–78, 115, 119, 121, 123–128, 132, 135, 136, 140, 142, 148–150
- multiple index files, 227
- multi-stream approach, 93, 94, 97–98, 105–107, 110
- multi-stream paradigm for ASR, 94. *See also* multivariable acoustic analysis
- multivariable acoustic analysis, 93. *See also* multi-stream paradigm for ASR
- N**
- natural language
- dialog, 38–42
 - interaction, 100
 - processing, 92, 93, 100, 110, 227, 252, 257, 268, 271–273, 281, 317, 319
- Natural language processing (NLP)
- discourse modeling, 317
 - part-of-speech tagging, 317
 - semantic inference, 317
 - syntactic parsing, 317
- Natural language understanding (NLU), 14, 208
- Naval Health Research Center (NHRC), 282, 283
- Naval Voice Interactive Device (NVID), 280
- navigation, 4, 5, 8, 11, 15, 19, 20, 23–25, 28, 29, 33, 36, 49, 50, 78, 84, 92, 101, 108, 109, 116–119, 122, 123, 125, 126, 140, 145, 146, 148, 150, 151, 192, 251, 269, 270, 293, 297
- navigation application, 23, 25, 58, 109, 293
- neonates, clinical abnormalities in, 344
- network connectivity, 19
- New England Journal of Medicine*, 265
- Nexidia’s ESP module, 226, 235
- Noise Harmonic Ratio (NHR), 315
- noisy channel conditions and dialectal variations, phonetic-based, 242
- noisy environments, 44, 110, 272
- noisy texts, 278
- non-speech sound, 158
- O**
- Objective measures
- hidden measures, 157, 158, 166–168, 171, 175
 - observable measures, 157, 163–166, 175
- off-line actions, transcription (annotation) as, 158
- offshore transcription industry, 249
- ongoing treatment, assessing progress of, 306
- open vocabulary, 228, 229
- open web search, 42, 49
- operator greediness, 158, 166
- opt-out (callers request human-agent assistance), 165–166
- Out of grammar, 37, 166, 167
- Out-of-vocabulary (OOV)
- rate, 68, 73–75, 224
 - words, 68, 278
- P**
- PACS, 253, 255, 262, 265, 270. *See also* radiology report
- patient care, quality of, 248, 251, 264, 270
- patient data, electronic management of, 277
- patient, privacy of, 319
- Performance Index (PI) function, 151
- personal devices, 4, 10, 15
- Personal Digital Assistant (PDA), 11, 31, 120, 251, 271
- personalization, 10, 33
- personal voicemail inbox, 15
- Pervasive Developmental Disorder Not Otherwise Specified (PDD-NOS), 312

- phonetic
phonetic context, 69
phonetic dictionary, 226, 229
phonetic index, 224–227, 229, 230, 232, 233, 235 (*see also* phonetic search track)
phonetic searching, 224–235
phonetic speech recognition, 21, 22
phonetic-based indexing and search, 222, 224, 230
phonetic search track, 224, 225
physically disabled users, 33
physioacoustic model, 325
pitch harmonics, 326, 329, 331, 332, 340, 342, 344
pitch rising, 342, 343. *See also* voicing
PocketSphinx engine, 104. *See also* Carnegie Mellon University (CMU)
preventive healthcare, 277, 285, 286, 300, 301
Principle Factors Analysis, 151
“print-ready” dictation, 251
probabilistic text mapping (PTM), 258
problem-oriented medical record (POMR), 263. *See also* Weed, Lawrence L.
Program for Evaluation and Analysis of all Kinds of Speech disorders (PEAKS), 316
Prosody-Voice Screening Profile (PVSP), 312
Psychiatric Content Analysis and Diagnosis system (PCAD), 309, 317. *See also* Latent Semantic Analysis (LSA)
Pyramid Research, 6
- Q**
quality performance, 277, 286, 293, 300
queueing theory, the psychology of, 183, 188, 189
- R**
radiology report, 253, 260–263
Real-Time Monitoring (RTM) application, 222, 239–240
real-time, speech analytic solutions that are robust to, 242
real-time speech recognition (SR), 252, 253, 255–257, 261, 273
Receiver Operating Characteristic (ROC) curve, 231
recognition error, 12, 37, 41, 44, 68, 78, 87, 148, 158, 192, 195. *See also* search errors
recognition hypothesis, 68, 157, 166. *See also* human transcription
record-keeping errors, 270
Relative Average Perturbation (RAP), 315
reliable parameters, extraction of, 93
repeat users, 62
repetitive stress injury, reduce the effects of, 248
re-prompts, number of, 151, 212. *See also* ASR accuracy
retry rate, 164–166. *See also* speech errors
RIS, 253, 255, 262, 272. *See also* radiology report
rule-induction algorithms C4.5, 307
Ripper, 307
support vector machines (SVM), 196, 197, 202
- S**
salience (emotional) value, measured by, 205–206
scalability, a level of, 230
schizophrenia, 305, 307, 311–312, 317
search
errors, 44
experience, multi-modality of, 65
request, 15, 68
speed, 230, 233, 234
by voice, 58, 61–89
searching dialog, 108
self service (self-service) applications for call centers, 181, 183
self-service path, abandoning of, 183
self-service system in the call center, speech-activated, 182–184
self-service transaction, 133
Semantic Language Models (SLMs), 185, 186
severely degraded environments, 93, 108, 110
Shapewriter, 28
shipboard environmental survey data, 280
Shipboard Non-tactical ADP Program (SNAP) Automated Medical System (SAMS), 281–283, 295, 296, 298
Signal-to-noise ratio (SNR), 92, 95, 105–108, 225
single-Gaussian context-independent systems, 69
single numeric value, 155
smart data strategy, 276, 300
smart handset, 125

- smart phone (smartphone)
 high end, 25 (*see also* iPhone, Blackberry, Android, Symbian, Windows Mobile Devices)
 Internet-enabled (web-enabled), 65
- Soldier's On-System Repair Tool (SPORT), 280
- speaker authentication, 4, 13
- speaker state, 305–319
- speaker state, statistical machine learning techniques for the study of, 306. *See also* Hidden Markov Model (HMM); rule-induction algorithms
- spectral analysis, 201, 226
- spectral and cepstral subtraction, 92
- spectrograms, 201, 226, 325, 330–333, 337, 339, 344, 345
- spectrographic analysis (spectrographic voiceprint analysis), 323–345
- speech-activated interface (speech-only interface)
 design of, 181–189
 to expand self-service over the phone, 184
 options, menu of, 189
 user expectation of, 183
 user satisfaction with, 182
- speech analytics, 12, 221–243
- speech content (word count and lexical patterns), 314, 317. *See also* speech signal
- speech-enabled intelligent agents, 110
- speech errors
 dis-confirmations, 165
 speech overflows, 165
 time-outs, 165
- speech recognition, adaptation by end-users, 276
- speech recognition capabilities and wearability, trade-offs between, 301
- speech signal
 durational features, 317
 F0, 317
 intensity, 317
- speech signals of a DSR system, extracting features from, 93. *See also* front-end process
- Speech Strategy News*, 5
- speech-to-text, 24–27, 34, 222, 230, 232, 251, 283
- speech-to-text dictionary, 222
- spelling mode, 101
- SPHINX-II recognizer, 101
- SPHINX, recognizers' family of, 101
- spoken audio information, quantitative intelligence from, 236
- spoken dialog (dialogue) systems
 deployed in call centers, 36, 156, 163, 164, 172
 interactive, 93
 on mobile communications, 93
 performance of, 159, 163
- spoken search, 63, 66. *See also* voice search
- spontaneous dialogue, ecological validity of, 318
- SR (speech recognition) dictionaries, language models with, 261
- standardized metric, for system performance, 177. *See also* Caller Experience Index
- structured
 clinical data, 268
 queries, 227–228
- structured numerical data, 148
- subjective measures
 caller cooperation, 159, 171, 174
 caller experience, 159, 171–175 (*see also* objective measures)
- Subject Matter Experts (call center agents), 116
- supervised learning, 69, 197. *See also* unsupervised learning
- surgical robot, 273
- Symbian, 25, 26, 33, 47
- system designer, 165, 177, 183, 185, 189, 278
- System Usability Scale (SUS), 55
- ## T
- TALKS, 23
- task and technology, a fit between, 277
- task-technology-fit (TTF) model, 276, 284–286, 291–293, 300, 301. *See also* smart data strategy
- text box, 14–15, 27, 34, 49, 52, 54, 55
- Text Retrieval Conference (TREC), 223
- Text-to-speech (TTS), 4, 12–13, 20, 22–28, 36, 41, 43, 120, 314
- The Nature of Technology: What It Is and How It Evolves* (W. Brian Arthur), 9
- TIDigits, 104, 105
- time-varying spectral characteristics, 330
- T9 Output *See also* T9Write, 28
- traditional clinical narrative, EMR systems threaten, 266. *See also* electronic patient narratives
- Transaction Completion Rate, 151
- Transaction Completion Time, 151
- Transaction Duration, 123. *See also* Average Handling Time (AHT)
- transformational modeling, designed for automated medical transcription, 258
- transmission channel, 93

transparency, value of from other real world self-service systems, 183, 186–187 triphone systems grown from decision trees, 69 triphone systems grown from decision trees and use GMMs with variable numbers of Gaussians per acoustic state, 69–70 True Confirm Total (*tct*), 167, 168 True Total (*tt*) and True Confirm Total (*tct*), 168 T9Write, 28

U

UIT search algorithm, 105 unconstrained mobile speech interface, 42–49 Universität Erlangen–Nürnberg, 316. *See also* Program for Evaluation and Analysis of all Kinds of Speech disorders (PEAKS) University of Southern California’s Keck School of Medicine, 262. *See also* radiology report unsupervised learning, 63, 69 unvoiced sound, 337 Usability improved, 10, 46, 65 testing, 53–54 tests, 53, 55 user-centered interfaces, 110 user experience improved, 109 multimodal, 116, 123, 150–153 user interface (UI) challenge to designer, 20 design of, 36, 38, 56, 62, 63, 76 effective, 62 experience, 17, 20 “say what you want” (SWYW), 13, 14 simple and intuitive, 14, 186 user profile, 93 Users emotional state of, 194, 197, 202, 209, 214 expectations, 39, 46, 61, 183 experience, 13–17, 20, 23, 30, 32, 43–46, 51, 53, 55, 56, 59, 67, 68, 76, 109, 116, 123, 124, 150–152, 172, 189 needs, 39–41, 44, 62, 116 Pragmatic Users, 32–34 satisfaction, 46, 85, 150, 151, 182, 279, 316 (*see also* CSAT indicators, preference score of) Social Users, 32, 33 studies, 62, 63, 85–89, 278, 284 Stylists, 33, 36

Technophiles, 33, 34, 36 user utterances (turns) angry, 196–203, 211–218, 310 non-angry, 197–203, 207, 211–215, 217, 218 U.S. Navy ships, medical end-users aboard, 281

V

Veterans Health Administration (VHA) EMR system, 266 Visual voicemail, 7 Vocollect Voice, 17 voice-activated, 22, 37, 38, 53, 121–123, 125, 130, 135, 182, 252, 261, 270, 275–301 Voice Activity Detection, 233 voice browser, 159, 161. *See also* application server voicemail-to-text services, 7, 15 voice search, 5, 24, 26, 58, 62, 65, 71, 73, 77–81, 85–89, 117, 121, 126, 135, 149, 150, 152 Voice Turbulence Index (VTI), 315 voice user interface (VUI), 14, 16, 17, 36, 47, 56, 116, 118, 122, 125–131, 170, 172. *See also* graphical user interface (GUI) Voicing, 337, 343–345 Vsuite product, 23 VUI “macro” commands, 135

W

wait (time) for callers, psychology of, 188, 189 wearable computing device, 281, 282, 301, 315 Web, navigating and searching of, 24, 33, 101, 104, 108 WebScore, a measure of sentence-level semantic accuracy, 70 web search, 13–15, 20, 24, 33, 42, 47–49, 52, 63, 65, 68, 78, 80, 85 Web search engine, 227 Weed, Lawrence L. (University of Vermont), 263. *See also* problem-oriented medical record (POMR) WIMP (windows, icons, mouse, pointer)-based machine, 109 wireless PDA devices, SR incorporated in, 251 Wizard of Oz experiment, 120 word error rate (WER), 67, 68, 73–75, 158, 177, 223, 224, 230, 278, 316. *See also* misrecognitions at word level wrapper approach, 124, 131, 136