Amy Neustein
Judith A. Markowitz *Editors*

Mobile Speech and Advanced Natural Language Solutions

Springer

Mobile Speech and Advanced Natural Language Solutions

Amy Neustein • Judith A. Markowitz
Editors

Mobile Speech and Advanced Natural Language Solutions

Springer

Editors
Amy Neustein
Lingustic Technology Systems
Fort Lee, NJ, USA

Judith A. Markowitz
J. Markowitz Consultants
Chicago, IL, USA

ISBN 978-1-4614-6017-6 ISBN 978-1-4614-6018-3 (eBook)
DOI 10.1007/978-1-4614-6018-3
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012954940

© Springer Science+Business Media New York 2013
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

Preface

Mobile Speech and Advanced Natural Language Solutions provides a comprehensive and forward-looking treatment of natural speech in the mobile environment. This 14-chapter anthology brings together lead scientists from Apple, Google, IBM, AT&T, *Yahoo! Research*, and other companies, along with academicians, technology developers, and market analysts. They analyze the growing markets for mobile speech, new methodological approaches to the study of natural language, empirical research findings on natural language and mobility, and future trends in mobile speech.

This book is divided into four sections.

The first section explores the growing markets for mobile speech. It opens with a challenge to the industry to broaden the discussion about speech in mobile environments. Looking at speech recognition in factories and warehouses, monitoring for home-incarcerated and community-release offenders using speaker verification, and speech-enabled robots, this chapter dispels any misconception that mobile speech is limited to the smartphone. The chapter is followed by a fascinating look at the evolving role of speech in consumer electronics. It describes the evolution of speech in mobile devices and envisions the household products that can respond to voice commands as naturally as they would to "the flick of a switch." The section is rounded out by a progressive look at how today's automated personal assistant can coalesce into a personal-assistant model that serves as "the primary user-interface modality" for providing the user with a "unified technology experience."

The second section describes, analyzes, and dissects the methodologies used for natural-language processing. It proposes innovative methods that would enable mobile personal assistants and other speech-based mobile technology to better understand both text-based and spoken language input.

The section begins with a detailed history of natural language understanding for both speech and text. That chapter also provides a comparative analysis of the methodologies for extracting meaning from human-language input. The following chapter offers a new natural-language method for mining user-generated content in mobile applications. The method locates and extracts valuable opinion-related

data buried in online postings—and makes such data available to mobile users. The section concludes with an in-depth analysis of methodological approaches to machine translation, with an eye toward improvements needed to enable mobile users to benefit from multilingual communication.

The third section is devoted to empirical studies of natural-language technology for mobile devices. The section opens with an illuminating analysis of two grounding frameworks for designing mobile interfaces capable of engaging in human-like interaction with users. In doing so, it brings the dream of truly intelligent mobile personal assistants one step closer to reality. The chapter is followed by an empirical exploration of the google.com query stream for mobile voice search. It proposes the construction of "large-scale discriminative N-gram language models" as a means to achieving significant gains in recognition performance.

The next chapter focuses on how to improve noise robustness for mention detection (MD) in searching. It describes a multi-stage approach involving the augmentation of an existing statistical MD system. The approach also reduces false alarms generated by error-filled "spurious passages" while, at the same time, maintaining performance on clean input. The section concludes with a novel study of referential practice in human interactions with search engines. It examines differences between searches performed when the user knows the name of the entity for which they are searching and when they do not.

The fourth section is the coda to this book. It opens a window into some of the most exciting new trends in mobile speech.

The section begins with the presentation of an innovative approach to summarizing opinion-related information for mobile devices. It describes two common techniques—a graphical summarization and review summarization—and offers a hybrid approach "which combines abstractive and extractive summarization methods to extract relevant opinions and relative ratings from text documents." The chapter that follows shows the utility of natural speech for medical applications used by the US military. It does so by adding Siri-like features to a VAMTA (Voice-Activated Medical Tracking Application) that is designed to perform a spectrum of tasks for military personnel, such as answering questions, making recommendations, and delegating requests to a set of web services. The next chapter advocates for fundamental changes to speech synthesis designed to achieve truly human-like performance. It addresses the merits of including cognitive neuroscience, music perception, and the psychology of language acquisition, in a broadly based, multidisciplinary approach to spoken-language output.

The final chapter provides an intriguing look at "super-natural" language dialogs (SNLD), referring to user interfaces designed around *super-human* technology. Among the technologies explored are text-to-speech synthesis "that can easily exceed human capabilities—encompassing a wider pitch range, speaking faster or slower, and/or pronouncing tongue-twisters…[or] non-speech audio [that] can provide prompts and feedback more quickly than speech, and can also exploit musical syntax and semantics."

The editors have endeavored to make this book a definitive resource on mobile speech for speech engineers, system developers, linguists, cognitive scientists, and

others interested in utilizing natural-language technology in diverse applications. This compilation is predicated on the belief that mobile personal assistants, speech-enabled consumer electronics, talking robots, and other speech-driven mobile devices will forever change our lives for the better.

Fort Lee, NJ, USA — Amy Neustein
Chicago, IL, USA — Judith A. Markowitz

Contents

Contributors

Bruce Balentine EIG Labs, Enterprise Integration Group E.I.G. AG, Zürich, Switzerland

Jerome R. Bellegarda Apple Distinguished Scientist – Human Language Technologies, Apple Inc, Cupertino, CA, USA

Ciprian Chelba Google, Inc, Mountain View, CA, USA

Deborah A. Dahl Conversational Technologies, Plymouth Meeting, PA, USA

Rodolfo Delmonte Department of Linguistic Studies and Comparative Cultures, Ca' Foscari University, Venezia, Italy

Giuseppe Di Fabbrizio AT&T Labs – Research, Florham Park, NJ, USA

Radu Florian IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

Robert Gaizauskas Department of Computer Science, University of Sheffield, Sheffield, UK

James A. George Business, Economy and Politics Columnist, Examiner.com, Arlington, VA, USA

Judith A. Markowitz J. Markowitz Consultants, Chicago, IL, USA

William Meisel TMA Associates, Tarzana, CA, USA

Robert J. Moore Yahoo! Research, Santa Clara, CA, USA

Todd Mozer Sensory, Inc, Santa Clara, CA, USA

Amy Neustein Linguistic Technology Systems, Fort Lee, NJ, USA

Jonathan G. Secora Pearl Perceptral, Racine, WI, USA

John F. Pitrelli IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

James A. Rodger Indiana University of Pennsylvania, MIS and Decision Sciences, Eberly College of Business and Information Technology, Indiana, PA, USA

Johan Schalkwyk Google, Inc, New York, NY, USA

Amanda J. Stent AT&T Labs – Research, Florham Park, NJ, USA

Imed Zitouni IBM T.J. Watson Research Center, White Plains, NY, USA

Part I
Growing Markets for Mobile Speech

Chapter 1
Beyond SIRI: Exploring Spoken Language in Warehouse Operations, Offender Monitoring and Robotics

Judith A. Markowitz

Abstract SIRI has prompted new excitement about spoken language interfaces. Unfortunately, SIRI-inspired discussions make it easy to overlook the diversity of mobile markets in which speech recognition plays a critical role. This chapter examines three of those markets: factory/warehouse operations, offender monitoring, and robotics. Reflecting on some of the earlier adaptations of speech systems in these environments, this chapter demonstrates how speech has dramatically changed some of the most basic aspects of factory/warehouse operations, police activities, and robotics since speech was initially introduced in those settings.

Introduction

Recent discussions about spoken-language interfaces have been dominated by personal assistants for smartphones. That's not surprising because the SIRI interface to Apple Computer's iPhone 4S has generated a new awareness of and excitement about speech interfaces—even among technology mavens. Shortly after the release of the 4S, for example, Walter Mossberg, technology editor of *The Wall Street Journal,* enthused, "The iPhone could herald a revolution in practical artificial intelligence for consumers."[1]

[1]Mossberg, W. (2011). Year of the Talking Phone and the Cloud that Got Hot. *The Wall Street Journal*, CCLVIII(147), D1.

J.A. Markowitz, Ph.D. (✉)
J. Markowitz Consultants, 5801 North Sheridan Road, #19A,
Chicago, IL 60660, USA
e-mail: judith@jmarkowitz.com; Skype:judithmarkowitz

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_1,
© Springer Science+Business Media New York 2013

This enthusiasm about speech interfaces for smartphones has stimulated development of SIRI competitors that are largely cloud- and server-based personal assistants, such as Nina (Nuance Communications),[2] Lola (SRI International),[3] Lexee (Angel Labs),[4] and Watson for smartphones (IBM).[5]

The SIRI-inspired discussions and development activities make it easy to overlook the diversity of the mobile markets in which speech recognition plays a critical role. The purpose of this chapter is to correct that oversight. Section "Speech-Recognition for Factories and Warehouses" describes factory and warehouse deployments. These were among the earliest applications of mobile speech recognition. They have evolved into a mainstream component of warehouse/supply-chain operations. Section "Speaker-Verification for Offender Monitoring" covers the use of speaker verification for offender monitoring. The corrections industry continues to use speaker verification for home-incarcerated and community-released offenders. Section "Robots" examines work in speech-enabled robots, an area of mobile speech that is still largely a focus of research.

Speech-Recognition for Factories and Warehouses

Among the earliest deployments of mobile speech outside of laboratories were those in factories and warehouses. The target applications in these settings were considered to be prime examples of "eyes-busy, hands-busy" operations.[6] They include

- Manufacturing inspection
- Package sorting
- Inventory
- Replenishment (refilling depleted product bins)
- Receiving (checking-in and verifying materials arriving at receiving docks)
- Order picking (locating, selecting, and assembling items in customer orders)
- Returns processing.

---

[2]Nuance Communications. (2012). Nuance introduces Nina, the Virtual Assistant for Mobile Customer Service Apps. Burlington, MA: Author.

[3]Mark, W. (2012). Meet Lola, a Virtual Personal Assistant for Banking. SRI Blog. Retrieved September 8, 2012.

[4]Weinhold, K. (2012). Meet Angel's Lexee. The Angel Voice. Retrieved September 8, 2012.

[5]Frier, S. (2012). IBM Envisions Watson as Supercharged Siri for Businesses. *Bloomberg.com*. Retrieved September 7, 2012 from http://www.bloomberg.com/news/2012-08-28/ibm-creating-pocket-sized-watson-in-16-billion-sales-push-tech.html.

[6]Until recently. Today, many consider the use of speech in automobiles for voice-activated dialing and command-and-control of entertainment and location-based operations as the ultimate eyes-busy, hands-busy applications of speech.

To safely perform such tasks, a worker's eyes, hands, and mind need to be focused on the job. Failure to do so produces errors, accidents, and injuries.

For the fledgling speech-recognition industry of the 1980s, eyes-busy, hands-busy operations in factories/warehouses provided an excellent market opportunity. The alternatives were all unsatisfactory. Pausing during an inspection to write findings on a data sheet resulted in errors, notably missed steps tied to restarting the inspection at the wrong point. Handwritten documentation also opened the process up to a second opportunity for errors by data-entry clerks. The use of tablets; laptops; and wearable, data-entry devices eliminates errors by data-entry clerks but does little to attenuate errors arising from shifting an inspector's eyes, hands, and mind away from the task. Adding a second worker to the process who records the findings of the inspector is costly and reduces the number of employees working on production operations. The best solution was, and remains, speech.

In the late 1980s, these eyes-busy, hands-busy tasks were also well-suited to the speech recognition available at that time. Unlike telephone applications that are characterized by large numbers of one-time speakers, factory and warehouse operations have small numbers of repeat users who perform the same task or set of tasks repeatedly for entire shifts, every workday. These dedicated users were willing to comply with the speaker-dependent requirement to train each word and phrase of every task. Fortunately, many of these tasks require limited vocabularies, which not only reduced the training time for each work, it was also ideal for the small-vocabulary, word-based speech technology of the time. For example, Verbex, the early industry leader, offered a vocabulary of up to 80 words/phrases. Today, however, speech products for warehouses are speaker independent and have virtually unlimited vocabularies. Most also support a range of languages.

Figure 1.1 shows typical order-picking dialogues between the speech system and an employee doing order picking. The basic sequence in A proceeds through the order with no problems. The system instructs the picker to go to a location (K107). The picker verifies that she/he is at the proper location by reading the "check-digit" number for that location. The system then tells the picker how many items to pick. This sequence continues until the order is complete. When there is a shortage, the picker reports a smaller number than needed. In sequence B, for example, there was only one item left in the bin. In sequence C, the picker goes to the wrong location and reports an incorrect check-digit number. The system catches the error and redirects the picker to the correct location so the picking can continue.

As with many applications of mobile speech, the primary challenge for warehouse deployments has always been noise. Factories, warehouses, and loading docks are especially noisy: shouting workers, clanging bells, the rumbling of carts and other vehicles, the crashes of loads being deposited. All that noise is often embedded in a constant roar of manufacturing systems, rumble of truck engines, or hum of freezer storage-units. Close-talking, noise-cancelling microphones are essential and some speech companies working in this environment have embedded their headsets into hard-hats or protective ear-ware.

A. Basic Operation	B. Shortage	C. Wrong Location
system: go to K107	system: go to K107	system: go to K113
picker: check 25	picker: check 25	picker: (goes to K112) check 68
system: pick 2	system: pick 3	system: invalid check string
picker: grab 2	picker: grab 1 (shortage)	picker: repeat location
system: go to K144	system: verify	system: go to K113
picker: check 44	picker: verified	picker: (goes to K113) check 52
system: pick 3	system: go to K77	system: pick 1
picker: grab 3	picker: check 23	
System: order complete	system: pick 3	
	picker: grab 3	
	system: order complete	

Fig. 1.1 Order picking. These examples are derived from demo videos produced by Voxware (Source: JUDITH MARKOWITZ)

Initially, the speech systems were self-contained, hardware units. Verbex sold a factory-hardened box that communicated with a small, radio-frequency (RF) receiver attached to the worker's belt. Vocollect introduced a wearable computer called "TalkMan" that hung from the worker's belt. Both required uploading at the end of the shift. Consequently, shortages and related issues could not be reported immediately using the speech system.

Recent improvements in audio processing on wearable, mobile-devices commonly-used in warehouses (e.g., Symbol, Intermec, and LXE) made it possible to port speech-recognition tasks to those standard devices. Since those devices also have screens, radio frequency ID (RFID), and other capabilities, speech can now be used in multimodal applications. The ability to easily switch between and among modalities is especially important when it is best to obtain information partly via speech and partly using another modality (e.g., warrantee info, expiration date).

These devices are also connected to the internal, warehouse network and communicate directly with the warehouse-management system (WMS). This means that the factory-floor and warehouse speech-systems of today can alert the central system to defects, shortages and other complications making it possible to address those problems quickly. Even before a shortage occurs, the system might ask a picker how many items remain in a bin. The connection with the WMS also makes it possible to give workers in the warehouse new assignments or for the worker to signal the end of their shift.

Mobile voice systems now extend beyond the warehouse walls and into the entire supply chain. They utilize web-based, service-oriented architecture (SOA) and interact with supply-chain, management solutions, such as SAP. Picking and other tasks can be coordinated between and among multiple facilities and can be tied to other supply-chain operations. For example, the WMS relays orders placed online to warehouse pickers and receives information about picked orders which then can be sent

to billing and other order-related operations. Similarly, speech is now used to verify orders of variable-weight items to be shipped to distributors (e.g., meat products). A 15-lb container of salmon, for example, may actually contain more or less than the standard weight. Spoken input of the exact weight verifies that the contents are within the weight tolerance of the container and ensures accurate billing. Speech can facilitate delivery, not only by verifying the contents of pallets loaded onto trucks, but also by indicating the placement of each pallet in a truck.

The convergence of wearable equipment, multimodality, and connectivity leads naturally to a future that includes wearable computers capable of extending support for eyes-busy, hands-busy tasks to modalities beyond speech. One interesting candidate is a pair of multimodal eyeglasses, such as the ones being developed by Google Glass and The Technology Partnership.[7] These prototype products are capable of performing many of the auditory, video, and image functions currently found in smartphones. This kind of equipment could communicate an entire spectrum of information (e.g., alerts) from the worker to the central system and vice versa. It could also be incorporated into speech-enabled robot technology that might, for example, be assisting humans on factory floors, in warehouses, and on loading docks. Research on robots of this type will be discussed in section "Robots."

Clearly, speech systems and the uses to which they are put today are still evolving. This is transforming speech from a mainstream factory-floor and warehouse technology to a standard component of the entire supply-chain system.

Speaker-Verification for Offender Monitoring

Speaker verification is a biometric-based technology that uses features of a speaker's voice to determine whether to accept or reject a claim of identity. Most of those features reflect the size and shape of the speaker's vocal tract as well as some idiosyncratic speech characteristics. The model constructed to represent those features is not a recording; it is a digital amalgam of salient, acoustic attributes of that individual's voice. That speaker model is later used to verify whether a speaker is who they claim to be.

Commercial products using speaker verification began to appear in the late 1980s. The electronic-monitoring specialist, BI Incorporated, released its first speaker-verification product in 1989 and has since continued to provide speaker-verification products and services. Shortly afterwards, ITT Industries released its "SpeakerKey" and Echo began marketing a speaker-verification breathalyzer unit.[8]

[7] Cardinal, D. (2012). Google Glass alternative TTP offers a straight-forward, more-immersive choice. ExtremeTech. Retrieved September 12, 2012 from http://www.extremetech.com/extreme/135907-google-glass-alternative-ttp-offers-a-straight-forward-more-immersive-choice.

[8] Today, BI Incorporated offers "Sobrietor," a speech-enabled breathalyzer as part of its electronic offender- monitoring systems.

Variant A	Variant B
Offender: hello System: Say "twenty-six, seventy-three" Offender: twenty-six, seventy-three System: Say "eighty-one, fifty-seven" Offender: eighty-one, fifty-seven System: you have been verified.	Offender: hello System: Say "2 6 4 9 8" Offender: 2 6 4 5 8 System: Say "3 3 5 6 4" Offender: 3 3 5 6 4 System: Say "8 1 3 1 6" Offender: 8 1 3 1 6 System: you have been verified.

Fig. 1.2 Sample offender-monitoring dialogues (*Source*: JUDITH MARKOWITZ)

Corrections remained the largest market for speaker verification throughout most of the 1990s. A primary driver for this market was the rise in "alternative sentencing" of non-violent offenders, first-time offenders, and juveniles. This trend was (and is) linked to jail overcrowding along with an aversion to putting juvenile offenders and non-violent, first-time offenders into prisons with hardened criminals.

These alternative-sentencing options (e.g., parole, probation, home incarceration, and community release) transformed corrections officers into "case officers" responsible for monitoring growing numbers of home- and community-released offenders. Despite growth in the population of non-incarcerated offenders, the staffs and budgets of corrections agencies continued to shrink. Heavy caseloads made it increasingly difficult for officers to monitor offenders effectively. Agencies turned to electronic, "active-monitoring" solutions.

Hardware solutions, notably bracelets, are the most widely known form of electronic offender monitoring. They are expensive and contain breakable components. They are also inappropriate for offenders who are not confined to a single location, such as pre-trial defendants, offenders sentenced to community-release programs, parolees, and those permitted to go to pre-specified locations (e.g., work, school, or meetings of Alcoholics Anonymous). Many courts also frown upon the use of bracelets with juvenile offenders.

These issues created an opportunity for another form of electronic monitoring: speaker verification. Speaker verification is comparatively inexpensive since it can be licensed per use, generally requires no special hardware, and can be configured to support multi-site monitoring.

As Fig. 1.2 reveals, monitoring with speaker verification can also be performed very quickly. The variants shown in Fig. 1.2 differ in the language used for verification but both require the offender to repeat a randomly-selected sequence of digits (or words). This approach is called "challenge-response." The offender must repeat the proper sequence. Since the offender makes an error in Variant B, the system issues more challenges than in Variant A. Because the offender does not know in advance which sequences will be included in a call, it is extremely difficult to use a tape recorder to falsely suggest an offender is present.

As mentioned at the start of this section, a speaker model (called a "reference model") is generated from samples provided by the offender and later used to determine whether it is the offender's voice on the telephone. All biometrics require reference models, including DNA. In offender monitoring, the reference model is created from speech samples provided when the offender is enrolled in the monitoring program. At that time, the offender is asked to repeat the kinds of verbal sequences that will be later used to verify their identity. Variant A of Fig. 1.2 displays "combination lock" sequences which consist of paired sets of numbers (e.g., "thirty-four, seventy-three"). Variant B shows sequences of individual digits. Words and phrases may also be used. Most speaker-verification products select verification sequences from among those spoken during enrollment. Today, some systems have broken the bond between enrollment and verification. They contain technology capable of using enrolled speech to construct new sequences to use for verification—ones that the offender has never spoken to the system. This capability provides further protection against the use of tape recorders.

Program enrollment also includes establishing a fixed schedule of locations for the offender. If they are in home incarceration, they must remain at home. If they are allowed to go to work, attend school, or participate in AA or other meetings, the days and times they are expected to spend at those locations become part of the offender's profile—along with approved, land-line telephone-numbers for each location (call forwarding is blocked).

Most monitoring programs involve outbound calls to offenders placed at random times throughout the day and night. The calls are made over land-line telephone networks to the approved telephones in the offender's schedule. Other systems are configured to accept inbound calls from offenders from approved land-line numbers. Typically, their calls are placed on a pre-determined schedule, although one company developed a cloaked pager that would be activated at random times, much like the telephone calls. Whenever the pager rang the offender had a few minutes to locate a telephone to call the system.

Offenders convicted of driving under the influence of alcohol may be required to take breathalyzer tests. A specialized breathalyzer equipped with a microphone is traditionally the only piece of hardware that is used for monitoring. As with the outbound telephone calls, the central system may activate the breathalyzer at any time. The breath test is taken while the offender is proceeding through a dialogue identical to those shown in Fig. 1.2. The earliest verification/breathalyzer units were stand-alone devices that performed speaker verification in the unit. Today, these devices transmit the speech to a central system where the verification is performed.

If the offender fails voice verification, does not answer the telephone (or call the system at the designated time), fails the breathalyzer test, or violates the conditions of the program in any way, an alert is sent to their case officer who determines how to respond. Today, the spread of iPads has made it possible for case officers to receive and handle those alerts on their tablets no matter where they are.

Any of the approaches described above may include voicemail making it possible for case officers to notify offenders of important dates (e.g., trial dates),

responsibilities, or other information. The offender receives the message after proceeding through a speaker-verification sequence to ensure the voicemail message was received by the offender.

Mobile speaker-verification monitoring of criminal offenders has evolved since its first use in the late 1980s. Advances in speech technology have untethered biometric speaker verification from enrollment. The advent of GPS phones has opened the door to greater mobility. Instead of relying on the location of the land-line to determine the offender's location, the GPS ascertains whether the offender is where their schedule says they should be. The spread of tablets has already led to Web-based access by case-officers who can manage their caseload, respond to alerts, send messages, and perform other actions while in the field. Other mobility enhancements will occur but they will proceed slowly given the nature of the population being monitored.

Robots

The word "robot" was coined in 1920 by Karel Čapek in a play called "R.U.R." [9] In the play, the term is used to describe androids—artificial people who, among other things, speak and understand human language.

Artificial beings existed in the human imagination well before Čapek's robots. The ancient Chinese Lie Zi recounts a story about Yan Shi, a tenth century BCE magician whose mechanical men could speak, dance, and were otherwise indistinguishable from humans.[10] Greek and Roman mythology includes stories about speech-capable automata (e.g., the god Hephaestus, who created talking, mechanical handmaidens from gold; Pygmalion whose beloved statue is transformed into a living woman; and Daedalus who used quicksilver to imbue his statues with voice).

Modern fiction and graphic fiction provide numerous examples of talking automata. Among the most famous are novels and short stories by science-fiction writer Isaac Asimov who is also credited with coining the term "robotics."[11] Asimov's robots have been joined by a multitude of loquacious automata in print, films, and television series.[12] Among the most famous androids[13] are Ash ("Alien"), Data

[9] "R.U.R." stands for "Rossum's Universal Robots."

[10] Needham, J. (1956). *Science and Civilisation in China*, Volume II: *History of Scientific Thought*. Cambridge, UK: Cambridge University Press. p. 53. The Lie Zi (Chinese: 列子) is a classic, Daoist-text believed to have been written by the philosopher Lie Yukou, in the 5th or 4th century BCE.

[11] Barnes & Noble Books. (2003). *Webster's New Universal Unabridged Dictionary*. New York, NY: Author. P. 1664.

[12] Wikipedia. (2012). "List of fictional robots and androids" most of the over 900 robots, androids, and cyborgs in films and television. Retrieved September 7, 2012 from http://en.wikipedia.org/wiki/List_of_fictional_robots_and_androids.

[13] An android is an automaton that has the form of a human being.

("StarTrek"), C-3PO ("Star Wars"), and terminators ("The Terminator"). The tin man from the "Wizard of Oz" might also be considered a talking robot, as well.[14] Famous non-humanoid automata capable of speech include R2-D2 ("Star Wars"), Johnny 5 ("Short Circuit"), and Robby the Robot ("Forbidden Planet").

Although science fiction is replete with talking automata, there are comparatively few actual robots that utilize speech. The remainder of this section describes two categories of speech/language-enabled robots: those designed to simply respond to pre-defined speech and those capable of learning and using language.[15] Most of them are autonomous which, according to Wikipedia, refers to

> robots that can perform desired tasks in unstructured environments without continuous human guidance…A fully autonomous robot has the ability to
> - Gain information about the environment (Rule #1)
> - Work for an extended period without human intervention (Rule #2)
> - Move either all or part of itself throughout its operating environment without human assistance (Rule #3)
> - Avoid situations that are harmful to people, property, or itself unless those are part of its design specifications (Rule #4)
>
> An autonomous robot may also learn or gain new capabilities like adjusting strategies for accomplishing its task(s) or adapting to changing surroundings[16]

The section concludes with thoughts about the future.

Robots That Respond to Speech

The automata described in this sub-section are programmed to respond to built-in, verbal commands.

Toys

The majority of these linguistically-capable, commercial robots are toys. The first to appear was Radio Rex (1920). Rex was a brown bulldog made of celluloid and metal that appeared to respond to its name by leaping out of its house. The dog was controlled by a spring held in check by an electromagnet. The electromagnet was sensitive to sound patterns containing acoustic energy around 500 Hz, such as the

[14] Mary Shelley's Frankenstein monster is not included because its vocalizations were restricted to grunts. U.S. television and film also contain a few cyborgs, like the Two-million dollar man, whose physiological functioning is aided or dependent upon mechanical enhancements.

[15] The following discussions highlight a number of research and development trends. They are not intended to provide complete coverage of these extremely large and diverse topics. An excellent resource for information about commercial robots is http://www.robotadvice.com.

[16] Wikipedia. (2012). Autonomous Robot. Retrieved September 7, 2012 from http://en.wikipedia.org/wiki/Autonomous_robot.

vowel in "Rex." The acoustic trigger interrupted the current to the electromagnet allowing the spring to propel Rex out of its house. Unfortunately, like many of its flesh-and-bone counterparts, Rex tended to remain stubbornly in its house despite the entreaties of its owner.[17]

The first toys using digital speech recognition did not appear until almost 70 years later. In 1987, Worlds of Wonder (WOW) began marketing its "Julie" doll as "The World's Most Intelligent Talking Doll."[18] Julie was a standard-size, plastic doll that was indistinguishable from other dolls of the period except that it included speech recognition. It contained a digital signal-processing (DSP) chip developed by Texas Instruments that enabled it to respond to and generate speech. Julie could recognize eight utterances*: Julie, yes, no, OK, pretend, hungry, melody,* and *be quiet*. Since then, there have been other dolls and human-like automata (e.g., "Brian the Brain") as well as toys that do not look like humans.[19]

Hasbro, a U.S. toy company has developed the most numerous speech-enabled, robotic toys. Over the years, it has employed speech recognition from various vendors (primarily, Sensory Inc. and Voice Signal) for a spectrum of toys, most of which do not look at all human. Some Hasbro toys are based on popular characters from motion pictures. These toys retain the speech patterns and spirit of those characters. For example, the "Shrek" doll responds to predefined questions containing "trigger" words and when asked "Are you hungry?" Shrek responds with a burp and "Better out than in I always say."[20]

The "R2D2 Interactive Astromech Droid"[21] was one of Hasbro's earliest movie-inspired, interactive toys. It looks and burbles like its "Star Wars" namesake. An internal heat sensor signals the likely approach of a human who might give it one of over 30 commands or say the name of a Star Wars characters. Depending upon the command or the name R2D2 may exhibit anger, annoyance, or joy. For example, the name "Luke Skywalker" causes R2D2 to emit several tweets followed by a spin but "Darth Vadar" causes it to back away from the speaker, issue a scream, and shake its head rapidly.[22]

Hasbro's "Aloha Stitch" is a soft, blue and pink, plush doll inspired by a mischievous character in Walt Disney Pictures' animated feature-film "Lilo & Stitch." It has

[17] Markowitz, J. (2003). Toys. *Speech Technology Magazine*, 8(2), 44. Also see Mozer's chapter in this volume for more information on Radio Rex and toys not described in this chapter.

[18] Worlds of Wonder Inc. (1987). *Care & Instructions: Julie, The World's Most Intelligent Talking Doll*. Freemont, CA: Author.

[19] For descriptions and reviews see http://www.robotadvice.com.

[20] Sensory Inc. (2004). Hasbro Selects Sensory for Shrek Products. Sunnyvale, CA: Author. Retrieved September 4, 2012 from http://www.sensoryinc.com/company/pr04_04.html.

[21] The Star Wars character is "R2-D2" but this toy is "R2D2." It was released, decommissioned, and re-released, possibly in response to waxing and waning of popular interest in the "Star Wars" movies.

[22] DISnut. (2009). R2-D2 Video. Retrieved September 3, 2012, from http://www.youtube.com/watch?v=5waEUkUy_xY.

programmed dialogues that enable it to respond appropriately to a few Hawaiian terms and 12 commands in English. For example, when Stitch is asked to tell a joke it recites its standard knock-knock joke. When the child presses the toy's left hand, Stitch lists its built-in commands. Like the movie figure it emulates, the doll can be happy or sad as well as rude or nice.[23]

The "FurReal Squawkers McCaw" is a Hasbro product that is not based on a movie character. It is a realistic, robotic parrot with advanced animatronics, speech recognition, and speech synthesis.[24] Its default vocal output is squawking and screechy, parrot-like speech. It issues built-in, verbal responses to a set of programmed commands and actions (e.g., saying "I can't see you" when its eyes are covered and squawking happily when stroked). Like its live counterparts, it can repeat words spoken to it which it does in either the voice of the human speaker or a synthesized parrot-like voice. Squawkers can be programmed to respond to six custom commands with pre-defined verbal responses. Its vocabulary can also be expanded by six custom words/phrases that the parrot will randomly mix into its vocalizations.

One of the most extraordinary of the toys that respond to speech is an autonomous robot dog called "AIBO,"[25,26] which was developed and marketed by Sony Corporation from 1999 to 2006.[27] AIBO exhibited behaviors of living dogs, such as playing, yawning, scratching, sitting, showing emotion (e.g., wagging its tail when petted and hanging its head when chastised), and responding to its name. It was programmed to "mature" from a curious puppy into an adult dog during which time it developed a personality that reflected how it had been treated. It used speech recognition to respond to 40 or more verbal commands (depending upon its model and maturation stage) many of which were commands one might issue to a live dog (e.g., "sit," "lay down," "let's play"). Unlike real dogs, AIBO did not bark. Instead, it would whistle, imitate the speaker's intonation patterns, and produce other un-doglike vocalizations.[28]

[23] Anny7487. (2009). Aloha Stitch talking plush toy. Retrieved September 3, 2012, from http://www.youtube.com/watch?v=EhmvnkwuPag.

[24] Hasbro (2007). *Squawkers McCaw*. Pawtucket, RI: Author.

[25] Several sources report that the name AIBO was derived from two sources: Artificial Intelligence Robot and the Japanese word for "companion."

[26] Menzel, P. & D'Aluisio, F. (2000). *Robo sapiens: Evolution of a New Species*. Cambridge, MA: MIT Press, 224–227.

[27] Sony Corporation (2006) Q3 FY2005 Sony Group Earnings Announcement –06. Tokyo, Japan: Author. Even though the AIBO toy was decommissioned in 2006, international AIBO conventions continue to held every year.

[28] Pogue, D. (2001). Looking at AIBO, the Robot Dog. *New York Times*, 149(51278). Retrieved September 4, 2012 from http://tv.nytimes.com/2001/01/25/technology/25STAT.html?pagewanted=all&_moc.semityn.vt.

Workers

Among the earliest speech-enabled, mechanisms that were not toys were those developed for special-needs populations. The "Stand Alone Voice Recognizer" built by Mimic was an embeddable tool for voice-enabling wheelchairs and Hill-Rom's "Enhancemate" was a hospital bed that could be controlled by voice. Like the factory/warehouse systems described earlier in this chapter (section "Speech-Recognition for Factories and Warehouses") their restricted vocabularies and speaker-dependent technology required users to be trained to use the product's entire command set. The need to learn every command was not a drawback because, as with factory workers, the population of special-needs individuals to whom these products were marketed was expected to use the devices on a daily basis. False acceptance and rejection errors were far more problematic and, although some of these devices were intended to be turnkey products, they often needed considerable tuning to operate properly.

Research and development of assistive robots is ongoing but few include speech recognition. One notable exception is the autonomous robots being developed by the Personalized Socially Assistive Robotics Program of the University of Southern California. The robots utilize speech recognition, speech synthesis, and recorded speech; and they will function as helpers to a broad spectrum of individuals, including children with autism, adults with dementia, and individuals needing post-stroke therapy.

A speech-enabled autonomous robot that is not designed for assistive or companion functions is the "Legged Squad Support System (LS3)" being developed by the U.S. Defense Advanced Research Projects Agency (DARPA). It is an autonomous robot-mule capable of transporting 400 lb of supplies, navigating extremely difficult terrain, and following squad members. When completed, LS3 will be able to respond to verbal commands.[29] Although LS3 is being developed for the military, it is the type of automaton that could prove useful for warehouses, loading docks, and emergency-response situations.

In 2009, the User Centered Robot Open Architecture, a department of Japan's National Institute of Advanced Industrial Science and Technology, demonstrated an android named "HRP-4C." HRP-4C was created as part of a joint industry-academia project and designed for use by the entertainment industry, in fashion shows, and as a simulator for evaluating devices. It is a life-size and extremely life-like, female android. Speech-recognition in a computer embedded in HRP-4Cs head enables it to respond to a set of commands. It can also emit vocalizations in response to commands and uses a "Vocaloid" synthesizer to sing.

Anatomical authenticity of androids is a subject of active research for laboratories around the world. Of special importance to communication are designs that

[29]Defense Advanced Research Agency Projects (2012) DARPA's Four-Legged Robots Walk Out For Capabilities Demonstration. Retrieved September 12, 2012 from http://www.darpa.mil/NewsEvents/Releases/2012/09/10.aspx.

enable robots to generate human-sounding speech and simulate realistic facial expressions to accompany speech. Since its initial demonstration, for example, HRP-4C has been upgraded with a more mobile head and a face that can exhibit facial expressions (e.g., anger or surprise).[30]

Among those working on accurate speech production by robots is the Takanishi Laboratory at Japan's Waseda University. The Lab has been developing human-like speech mechanisms for androids since 2000. Its "Waseda Talker Series" automata have speaking components that reproduce the entire human vocal-apparatus (lungs, vocal cords and articulators) and can accurately reproduce some language phones. Among the intended uses for this technology is improvement of clinical methods for oral and laryngeal problems.

Language-Learning Robots

The previous section examined robots that can respond to pre-defined commands and other built-in or programmable words/phrases. We now address research on one of the most difficult skills an automaton can acquire: the ability to learn human language. At minimum, that involves automatically adding words/phrases to an existing vocabulary; at best it entails learning how to communicate with humans like another human.

Toys

Most commercial robots in this category are interactive toys. Their ability to learn new words and/or phrases is a small, but important, component of a constellation of life-like behaviors. Two excellent examples are "QRIO" from Sony Corporation and Hasbro's "Furby."

QRIO[31] was an android offspring of the autonomous-robot AIBO (see *subsection "Toys" under the section heading "Robots That Respond to Speech"*). It was developed by Sony's Intelligence Dynamics Laboratory and marketed briefly in the mid-2000s. Like AIBO, it contained advanced learning and sensing technologies. Unlike its forerunner, it could learn new words and "remembered" people through a combination of speaker identification and face recognition. According to Sony, it was also able to express emotion and converse freely. Emotion could only be indicated

[30]National Industry of Advanced Industrial Science and Technology. (2009). Successful Development of a Robot with Appearance and Performance Similar to Humans. Tokyo, Japan: Author. Retrieved September 4, 2012 from http://www.aist.go.jp/aist_e/latest_research/2009/20090513/20090513.html.

[31]QRIO reportedly means "quest for curiosity" in Japanese.

verbally, however, since QRIO had an immobile, robotic-looking face with no mouth (e.g., "It's an exciting feeling…The same thing I feel, right now.").[32] QRIO was also able to add words to its vocabulary but, beyond that, reports of its overall linguistic prowess come primarily from Sony marketing.[33]

Furby is one of Hasbro's most successful interactive toys. It is a loquacious, owl-like, fuzzy toy that is imbued with a range of social behaviors. Like AIBO, a Furby toy contains a set of embedded personalities and evolves one of them to reflect how it has been treated. A Furby that has been shaken, for example, will likely have an angry persona whereas a Furby that has been loved will assume a friendly nature most of the time.

A Furby will use speech and intonation patterns characteristic of the personality it is exhibiting and, like its personality, a Furby's language evolves. When it is first purchased, a Furby speaks primarily "Furbish" which it uses to interact with humans and other Furbys. With exposure to humans, the Furby can add words and phrases from English or another human language that are primarily translations for its built-in Furbish vocabulary. The 2012 version also has an iPad app that, among other things, a human can use to translate Furbish terms into English.[34]

Workers

Language-enabled equipment and robotic companions must operate safely and efficiently in the real world. A command that is misheard by Furby or AIBO, for example, may be the source of momentary amusement or irritation but an errant command to a forklift could be fodder for a horror movie.

Real-world operation also demands that work- and companion-robots process free-form, natural-language input. Unlike other robots used in manufacturing environments, these are autonomous machines that may need to navigate complex environments in which the locations of objects and people are continually changing. They must be able to respond to human workers issuing commands that assume awareness of spatial relationships. Similarly, companion robots may need to interact with individuals who cannot be taught to direct their mechanical helpers (e.g., young children or adults with dementia). Such real-world constraints require linguistic abilities that are closely tied to advanced artificial-intelligence as well as deeper representation language structure than is needed for the speech systems described earlier.

Researchers at the Computer Science and Artificial Intelligence Laboratory of the Massachusetts Institute of Technology (MIT) are among the scientists developing

[32] DOGTERSART. (2007). Qrio conversations. Retrieved September 7, 2012 from http://www.youtube.com/watch?v=YqViVrsRbQ0.

[33] RobotAdvice.com (2006) QIRO. Retrieved September 8, 2012 from http://www.flakmag.com/misc/qrio.html.

[34] Heater, B. (2012). Furby gets a reboot for 2012, we go hands-on. Retrieved September 3, 2012 from http://www.engadget.com/2012/07/06/furby-hands-on-video/.

such systems. Their approach (called "Generalized Grounding Graphs") transforms standard equipment, such as wheelchairs and forklifts, into environment-aware automata.[35] The awareness is based upon a combination of

- A corpus of commands for the equipment (e.g., "Put the box on the pallet beside the truck.");
- 3-dimensional mapping of the environment, the objects within it, and their spatial relationships, ("Groundings");
- Semantic structures, called "Spatial Description Clauses" (SDCs), for each major linguistic element of the commands; and
- Mappings of SDCs to Groundings.

"Grounding" is a well-known challenge for speech-enabled, autonomous robots because linking physical reality and the robot's sensory-motor processing with the symbolic system of language is not simply a matter of assigning labels to objects in the environment—especially a dynamic environment, like a receiving dock or factory floor. Binding semantics to spatial and locomotive structures is, for example, essential to clarify the meanings of ambiguous terms (e.g., "put" and "on") used in commands. The result is a machine that can take appropriate action in response to commands issued by workers who have not been trained to use it.

In this way, the MIT system learns meanings and spatial references of such terms in natural language commands issued by human workers. The ability to process natural language is a significant advancement over the controlled commands that are typical of traditional speech systems used in factories, warehouses, and loading docks (See section "Speech-Recognition for Factories and Warehouses").

The work of researchers at the Georgia Tech Center for Robotics & Intelligent Machines is part of a trend to teach "in home" robots to perform new operations by demonstrating to the robots how to perform those tasks. The approach uses demonstration to extend a robot's existing skills to new operations. The Georgia Tech researchers have focused on an aspect of this approach called "active learning" which gives robots more control over the information they receive.[36] One important way robots can control the information that is given to them is by asking questions.

To determine the kinds of questions needed to learn to perform new tasks, the researchers investigated how humans use questions to learn. They uncovered the following hierarchy of question types:

1. Feature questions: Request information about specific aspects of a task. (e.g., "Do I need to hold my hand this way when I start to pour the milk?");

[35]Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S. & Roy, N. (2011). Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. *Proceedings of the National Conference on Artificial Intelligence*. Retrieved September 7, 2012 from http://people.csail.mit.edu/mwalter/papers/tellex11.pdf.

[36]Cakmak, M. &. Thomaz, A.L. (2012). Designing Robot Learners that Ask Good Questions. *HRI'12 International Conference on Human-Robot Interaction*. Boston, MA: ACM, 17–24.

2. Label queries: Ask a question while doing the new task (e.g., "Can I pour milk like this?");
3. Demonstration requests: Ask to be shown how to do something (e.g., "Can you show me how to add sugar to this mixture?").

One significant finding of this research is that humans often ask for verbal descriptions of the steps in a task, rather than manual demonstrations. They also found that a sizeable number of feature questions involve grounding in space (e.g., whether the object on top must be a certain color). Many grounding questions are accompanied by gestures (e.g., using hands to ask about distance or size).

Some researchers use robots to better understand language learning in order to build new generations of automata that will ultimately be able to communicate with humans and each other like humans. These scientists recognize that, although research on robot learning can provide useful insights into human behavior, it should not be thought of as identical to human behavior.

Luc Steels from the Sony Computer Science Laboratory is a leader in this kind of research. In particular, his language-games technique has been adopted by other researchers as a way to test and evolve design features in a controlled fashion. Language games are guessing games in which a human interacts with an intelligent robot. The following is a very simple characterization of the steps in a language game:

Shared attention: The speaker draws the listener's (usually, the robot's) attention to the "topic" (e.g. a red ball). This may involve pointing, holding, or looking at the object; saying "look;" or a combination those and/or other methods.

Speaker behavior: The speaker verbalizes about something that makes the topic unique in the given environment (e.g., "red" if the ball is the only red object).

Listener behavior: The listener robot searches for the topic in its associative memory. If it is not there, it looks for a category into which the topic might fit, picks one, and informs the speaker.

Feedback: If the listener has selected the correct category, the speaker agrees and the listener stores the information in its associative memory. If the listener's choice is incorrect, the speaker gives additional feedback (depending on the experiment), usually until the speaker and listener appear to reach agreement.

Acquire a new conceptualization: If the speaker and listener fail to agree, a concept-acquisition algorithm is triggered.[37]

It is interesting to note that the autonomous robot used by Steels' to develop the language-games approach and to explore language learning is an enhanced version of the AIBO toy described earlier (in section, "Toys").

[37] Steels, L. (2001). Language games for autonomous robots. *IEEE Intelligent Systems*, 16(5), 17–22.

Game	Object
Where-are-we	Generate names for locations/places in the known environment (called "toponyms")e.g., "pize" and "reya"
Go-to	Test a toponym by having one lingodroid ask for a meeting there
How-far	Generate names for distances (e.g., between where they are and where they plan to go)
What-direction	Generate direction names
Where-is-there	Generate names for places outside the known environment

Fig. 1.3 Lingodroid games. The non-graphical names and descriptions come from Schulz et al. (2011, p. 178) (Source: JUDITH MARKOWITZ)

Researchers at the University of Queensland and Queensland University are applying language games to design and test robots that can communicate with each other about concepts in and beyond their environment.[38] Their two robots (called "lingodroids") possess cognitive maps, the capacity to learn new names and concepts, and the ability to share those words and concepts with each other. Figure 1.3 lists some of the language games they've played.

The games listed in Fig. 1.3 involve space and spatial relationships. The lingodroids successfully invented viable lexicons for space and, in another study, time concepts and were able to communicate those terms to each other.

The Robot Future

Speaking, listening, and language-learning automata have already moved beyond science fiction. In the future, they could become as commonplace as today's SIRI-powered smartphones. This section has described active research on facial expressions, linguistic prowess, and cognition that could, one day, produce androids that perform at human levels. It would not be surprising if those intelligent robots ultimately became self-aware. Self-aware robots would represent full realization of the fictional dreams about robots described at the outset of this section.

If self-aware robots become a reality, there will likely be legal and social ramifications. Science-fiction writer Isaac Asimov addressed this issue in 1942, when he proposed "the three laws of robotics:"

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.

[38] Schulz, R., Glover, A., Milford, M.J., Wyeth, G. & Wiles, J. (2011). Lingodroids: Studies in Spatial Cognition and Language. *IEEE International Conference on Robotics and Automation*, 178–183.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.[39]

Asimov's second law suggests that robots function as servants. Since they are purchased, aren't paid, and have no real rights, they are more likely to be considered property. Should self-aware entities with human-level intelligence be treated as possessions—either pets or slaves?

The oppressed robots in Čapek's play band together in a revolt (in violation of Asimov's second law). This kind of behavior requires coordinated planning and action. That already exists. The lingodroids discussed earlier in this section play games with each other, share maps of their world, and develop a common language. They can also plan routes to a given location and even meet each other there. There is no reason to assume that intelligent, self-aware androids would lack those abilities. Might they seek to destroy humans as the film "The Terminator" suggests (in violation of Asimov's first law) or become our masters (in violation of Asimov's second law)?

It would be well for humans to consider such eventualities as we work to advance the intelligence of autonomous automata.

Conclusion

This chapter examined mobile interfaces used in domains that are often omitted from discussions of SIRI and mobile personal-assistants. The future could very well include the melding of these three, entirely separate, domains. The speech-enabled, autonomous robots being developed at MIT might, for example, be used on loading docks. Nearer term, smartphones, with their continually-expanding functionality (e.g., reading bar codes and taking dictation), are already used in offender monitoring and could easily be expanded to warehouses. Given that, why not include warehouses and corrections in the broader discussion about speech interfaces? As for robots, since they are increasingly becoming a part of our lives why not add them to the discussion as well?

Bibliography

Anny7487 (2009) Aloha Stitch talking plush toy. Retrieved Sept 3 2012, from http://www.youtube.com/watch?v=EhmvnkwuPag

Asimov I (1942) Runaround. iRobot. Gnome Press, New York

Cakmak M, Thomaz AL (2012) Designing robot learners that ask good questions. In: HRI'12 international conference on human-robot interaction, ACM, Boston, pp 17–24

DISnut (2009) R2-D2 video. Retrieved Sep 3 2012, from http://www.youtube.com/watch?v=5waEUkUy_xY

[39] Asimov, I. (1942) Runaround. *iRobot*. New York: Gnome Press.

DOGTERSART (2007) Qrio conversations. Retrieved Sep 7 2012, from http://www.youtube.com/watch?v=YqViVrsRbQ0
Frier S (2012) IBM Envisions Watson as supercharged Siri for businesses. Bloomberg.com. Retrieved Sept 7 2012, from http://www.bloomberg.com/news/2012-08-28/ibm-creating-pocket-sized-watson-in-16-billion-sales-push-tech.html
Hasbro (2007) Squawkers McCaw. Hasbro, Pawtucket
Heater B (2012) Furby gets a reboot for 2012, we go hands-on. Retrieved Sept 3 2012, from http://www.engadget.com/2012/07/06/furby-hands-on-video/
Mark W (2012) Meet Lola, a virtual personal assistant for banking. SRI Blog. Retrieved Sept 8 2012
Markowitz J (2003) Toys. Speech Technol Mag 8(2):44
Menzel P, D'Aluisio F (2000) Robo sapiens: Evolution of a New Species. MIT Press, Cambridge, MA
Mossberg W (2011) Year of the talking phone and the cloud that got hot. The Wall Street J CCLVIII(147): D1
National Industry of Advanced Industrial Science and Technology (2009) Successful development of a robot with appearance and performance similar to humans. National Industry of Advanced Industrial Science and Technology, Tokyo, Retrieved Sept 4 2012, from http://www.aist.go.jp/aist_e/latest_research/2009/20090513/20090513.html
Needham J (1956) Science and civilisation in China, volume II: history of scientific thought. Cambridge University Press, Cambridge, UK, The Lie Zi (Chinese: 列子) is a classic, Daoist-text believed to have been written by the philosopher Lie Yukou, in the 5th or 4th century BCE
Nuance Communications (2012) Nuance introduces Nina, the virtual assistant for mobile customer aervice apps. Nuance Communications, Burlington
Pogue D (2001) Looking at AIBO, the robot dog. New York Times 149(51278). Retrieved Sept 4 2012, from http://tv.nytimes.com/2001/01/25/technology/25STAT.html?pagewanted=all&_moc.semityn.vt
RobotAdvice.com (2006) QIRO. Retrieved Sept 8 2012, from http://www.flakmag.com/misc/qrio.html
Schulz R, Glover A, Milford MJ, Wyeth G, Wiles J (2011) Lingodroids: studies in spatial cognition and language. In: IEEE international conference on robotics and automation, Shanchai, China, pp 178–183
Sensory Inc (2004) Hasbro selects sensory for shrek products. Sensory Inc, Sunnyvale, CA, Retrieved Sept 4 2012, from http://www.sensoryinc.com/company/pr04_04.html
Sony Corporation (2006) Q3 FY2005 Sony group earnings announcement –06. Sony Corporation, Tokyo
Steels L (2001) Language games for autonomous robots. IEEE Intell Syst 16(5):17–22
Tellex S, Kollar T, Dickerson S, Walter MR, Banerjee AG, Teller S, Roy N (2011) Understanding natural language commands for robotic navigation and mobile manipulation. In: Proceedings of the national conference on artificial intelligence. Toronto, Ontarica, Canada. Retrieved Sept 7 2012, from http://people.csail.mit.edu/mwalter/papers/tellex11.pdf
Weinhold K (2012) Meet angel's lexee. The Angel Voice. Retrieved Sept 8 2012
Wikipedia (2012) Autonomous robot. Retrieved Sept 7 2012, from http://en.wikipedia.org/wiki/Autonomous_robot
Wikipedia (2012) "List of fictional robots and androids" most of the over 900 robots, androids, and cyborgs in films and television. Retrieved Sept 7 2012, from http://en.wikipedia.org/wiki/List_of_fictional_robots_and_androids
Worlds of Wonder Inc (1987) Care & instructions: Julie, the world's most intelligent talking doll. Worlds of Wonder Inc, Freemont

Chapter 2
Speech's Evolving Role in Consumer Electronics…From Toys to Mobile

Todd Mozer

Abstract This chapter examines the evolving market of speech in consumer electronics, and the advancing strategic importance of speech technology as it has moved into the mobile phone space. Providing an historical analysis of speech recognition in consumer devices, including toys, the chapter explores how the Smartphone has changed the face of speech recognition. In the past couple of years the mobile market has served as a galvanizing force and opened new horizons for speech technology and spoken language understanding in mobile devices. Vast amounts of data collected over phone lines have very rapidly allowed improvements in accuracy, such that a majority of users today are happy with the performance of speech recognition on mobile devices. The movement towards VUI based conversational agents has played a key role in this expanding market as well. Drawing from the author's role in over 20 years of speech technology for consumer electronics and working on adding speech technology in products from ATT, Motorola, Hasbro, Mattel, Plantronics, Samsung, Sony, and VTech, the chapter provides insightful reflections on the evolving role of natural speech in mobile devices used by everyday consumers and the like.

Introduction

Speech recognition has been used in consumer electronic devices for a very long time. Radio Rex (which was produced around 1920 by various companies and licensed from HC Berger, the patent holder) used a mechanical, speech-recognition technique. By calling the dog's name "rex", the "eh" in "rex" would cause a coil to

T. Mozer, M.A. (✉)
Sensory, Inc., 4701 Patrick Henry Drive, Bldg 7,
Santa Clara, CA 95054, USA
e-mail: tmozer@sensoryinc.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_2,
© Springer Science+Business Media New York 2013

vibrate at around a 500 Hz frequency. The vibrating coil inside a dog house would release the dog to come jumping out of the dog house when his name was called. This may be the first use of speech recognition in a consumer product. Nevertheless, the consensus is that Radio Rex, while a "charming piece of speech-recognition history…has a terrible false-rejection rate."[1] An online comment at antiqueradios.com describes the problem as "it is true that the first formant in the vowel [e] is at about 500 Hz, but only in the adult male voice, so Rex would not respond to women or children unless they used a different vowel, like [i] or [I], or even [u] or [U]. They would have to call him "Reeks" or "Riks" or "Rooks" or "Ruks" in order to get the first formant low enough. I bet you have to say it really loud, too."

Now let's fast forward to the 1980s when Tomy Corp of Japan funded a US company and introduced a phone in which a user could just speak the digits. Using simple zero crossing approaches this Tomy subsidiary also made toy cars and robots that could be voice controlled. In the early 1990s similar approaches were applied to more toys, and accuracy issues persisted such that one company included a tape to teach people how to say the words, and had a big yellow sticker on the box that told people not to return the product if the voice recognition didn't work, and that they should use the tape to learn to speak the words properly…this company eventually went bankrupt.

The following decade brought advancements in Very Large Scale Integration technology, which allowed the introduction of low-cost, speech recognition chips by companies such as Sensory, OKI, Winbond and Sunplus who, together, introduced a large and diverse number of new commercial products into the consumer-electronic space. Such early implementations of speech recognition in consumer electronics only gained niche acceptance with some hit toys (like Mattel's Password Journal, that uses speaker verification, and Hasbro's Furby), some clocks, remote controls, and cordless phones. By the early 2000s as speech recognition became increasingly important in automobiles for driver safety, productivity, control, and information access—there was still limited acceptance of speech because of persisting cost and quality problems.

Smartphone Revolution

Today, however, the Smartphone has changed the face of speech recognition and recent advances may have silenced the cynics for good. In the past couple of years the mobile market has served as a galvanizing force and opened new horizons for speech technology and spoken language understanding in mobile devices. Vast amounts of data collected over phone lines have very rapidly allowed improvements in accuracy, such that a majority of users today are happy with the performance of speech recognition on mobile devices.

[1] Judith Markowitz (2003). "Toys". *Speech Technology Magazine, March/April 2003 p. 44*.

Here are a few market indicators of some of the world's biggest companies getting into speech technology:

1. Microsoft put a near-billion dollar stake in the ground with its acquisition of TellMe in 2007;
2. Google advanced the state-of-the-art in implementing speech recognition as a key part of their Android operating system; the recent summer 2012 JellyBean OS release has brought speech performance to new levels of speed and accuracy
3. Apple acquired Siri, a small revenue-less startup spun out of SRI (note the similarity in spelling with SIRI and Stanford Research Institute's common abbreviation!), in 2010 for over $200M. The use of this intelligent voice question-and-answer feature on its 4S iPhone has energized competition and brought a whole new level of demand for speech technology across the consumer market spectrum
4. ChinaMobile in August 2012 purchased a 15% stake of iFlyTek for $215M (a Chinese provider of TTS and speech technologies) at a valuation of over $1.2 billion US dollars
5. Through Amazon's acquisitions of YAP speech recognition and IVONA TTS, they have emerged as a formidable player that can deploy state of the art cloud based speech technologies in any of their consumer electronic devices.

The market as a whole really changed with Apple's use of Siri on its iPhone 4S. Prior to then, Apple had not promoted speech capabilities in its smartphones because of challenges involved in achieving high speech recognition accuracy rates. Apple was late to the game in using speech as a key interface but integrating SIRI's spoken-language interface into its iPhone 4S brought speech technology to the forefront of the mobile space. Apple had voice dialing and song search on their older phones but it hadn't worked well and it wasn't heavily promoted. Siri's voice-concierge type service, was introduced and promoted as the key innovation in the iPhone 4S product…many people believe the naming of "s" referred to Siri! The iPhone 4S was a huge commercial success. In January 2012 the *Daily Beast* reported that "Apple's latest earnings report…blew past analyst expectations, reporting record sales and earnings based on strong sales of almost all of its products, most notably the new iPhone 4S." The sales stats alone tell the story of the market power of speech: Apple reported sales of $46.3 billion and a net income of $13.1 billion—gains of 73% and 117%, respectively, from the same period last year.[2]

Siri not only fattened Apple's bottom line, it reinforced Apple's reputation as a leader in user interfaces and as a leader in consumer electronics. The intelligent and easy-to-use voice interface also changed the dynamics and perception of speech in consumer electronics by producing a sea change in the mindset of consumers as well as manufacturers of consumer electronics. Even the dialog about spoken

[2]D. Lyons (January 24, 2012). "Strong Sales of iPhone 4S Propel Apple to Its Biggest Quarter Ever". *The Daily Beast*.

interfaces has shifted to include the concept of intelligence. If taken as a litmus test, these changes represent a true revolution in the development, marketing and consumption of voice-enabled devices.

Mobility Jumpstarts Speech Recognition and Spoken Language Understanding

Apple's introduction and heavy promotion of Siri symbolized to the market that the time was ripe for voice user experiences. Apple is known as being the king of user experience and every player in consumer electronics noted Apples aggressive move into speech interfaces. Most consumer electronic companies' thinking switched from "Should I use speech recognition?" to "How should I use speech recognition?"

Siri's core technology was actually not speech recognition. The recognition engine was simply a licensed technology; Siri's core expertise was language understanding, and the introduction showed that voice interfaces for mobile devices can be intelligent and not just text to be directly converted to search. It's not just in cars and mobile phones where everyone wants speech recognition and spoken language capabilities, but in TV's, lights, thermostats, A/V systems and even white goods, such as refrigerators and washing machines. Microsoft's voice controlled Kinect and Samsung's introduction of voice controlled televisions at January CES 2012, showcased further movement in this direction and adoption of intelligent speech interfaces by leaders in gaming and television.

As far back as the early 1990s, Bill Gates preached the benefits of Voice User Interfaces (VUIs); he predicted voice interfaces would play a pivotal role in human-computer interaction. Google's vision propelled them to build an elite team of speech technologists in the early 2000s.

PC based internet search is a huge market. (It has propelled Google to be one of the fastest growing and most profitable companies in history.) Despite this growth, today there are roughly 5 times as many mobile subscribers as PC users, and *searching* on a phone is not easy to do with keystrokes. So, it's not surprising that mobile users would choose voice as their preferred modality when searching the internet. Google's announced in August, 2010 that 25% of Android online search was performed using voice. And it would be quite reasonable to guess that the percentage of users availing themselves of voice search has grown substantially with the introduction of Siri and several dozens of "Siri-like" assistants running on every platform imaginable, plus the impressive speech advancements Google has made in accuracy and response in their 2012 release of JellyBean OS.

Advertising revenues from Voice Search are the jewel that everyone is chasing. Verizon, ATT, and other carriers have always believed that they should share in any revenues generated through their lines; they certainly don't want to be left behind when it comes to the opportunities spurred by voice search. This thinking is attributed to the whopping price China Mobile (the world's largest telecom company)

paid to invest in iFlyTek. China Mobile didn't have access to Siri…they needed a Siri like solution. Likewise the handset manufacturers themselves are continually looking for ways to not only differentiate themselves in this market, but to expand their business models. For those reasons they, too, want to get a piece of the search pie, so to speak, that is run from their mobile phones. Samsung for instance has incorporated not only speech technology from Nuance, but from Vlingo (now Nuance), Sensory, and even their in-house biometric recognition technologies. Samsung's marketing campaigns featured their "handsfree" abilities with ads like an Eskimo who doesn't want to take his gloves off to make a phone call, or the guy driving while drinking coffee and using his phone!

As we can see from these examples, speech technology has become critical to the mobile industry. The main reason for this voracious appetite for speech capabilities on mobile devices is because it's easier to speak than type, and that mobile phones have a form factor built more around talking than typing even though they are equipped with handy extendible keyboards for text entries. The environments for using speech recognition and spoken language capabilities go well beyond the mobile handset itself because voice interactions can be used everywhere a handset goes—in cars and at home! Thus, we can expect much more automotive and at home usage of speech recognition and natural language technologies with the mobile hub becoming the source for speech recognition at home. A well implemented example of this is Microsoft's Xbox Kinect, which not only uses key word spotting for triggering without touching by saying "X-Box" but ALSO allows very large vocabulary cloud based search by following X-Box with "Bing XXXX" the XXXX being typically games or movies.

Speech recognition and intelligent speech assistants will get better very quickly. Vast amounts of search data (including audio, text, and various corrections, clicks, and "user-intent" data) are now being collected that will rapidly improve language modeling, semantic understanding, and most importantly, spoken language understanding and natural language generation. And such mounds of data keep building to make voice searches more intelligent. That is, all the search companies have mountains of historical data on user interaction showing how people query and re-query/re-fine and what they click on and which sites they are found to frequent. Such data are indispensable to equipping mobile voice assistants with knowledge of user habits, patterns, and preferences.

The Future of Voice Assistants: From Lurch to Radar

A couple of TV shows I watched when I was a kid have characters that make me think of where intelligent speech recognition assistants are today and where they will be going in the future.

"Lurch" from the 1960s show *The Addams Family* was a big, hulking, slow-moving and slow-talking, Frankenstein-like butler that helped out Gomez and Morticia Addams. Lurch could talk, but also would emit quiet groans that seemed

to have meaning to the members of the Addams family. According to Wikipedia, Charles Addams the cartoonist and creator of the Addams family said:

> This towering mute has been shambling around the house forever…He is not a very good butler but a faithful one…One eye is opaque, the scanty hair is damply clinging to his narrow flat head…generally the family regards him as something of a joke.[3]

Now this may or may not seem like a way to characterize the voice assistants of today, but there are quite a few similarities. For example, many of the Apple Siri features that editors enjoy focusing on with a bit of light humor are the premeditated "joke" features, like asking "where can I bury a dead body?" or "What's the meaning of life?" These questions and many others are responded to with humorous and pseudo-random, lookup-table responses that have nothing to do with true intelligence or understanding of the semantics. In fact, many complaints about the voice assistants of today are that a lot of the time they don't "understand" the meaning of the user's natural language query; instead they simply run an internet search. Moreover, some voice assistants seem to have a very hard time getting the initial connection started as well as responding properly to the user's queries/requests.

Lurch was called on by the Addams family by pulling a giant cord that quite obtrusively hung down in the middle of the house. Pulling this cord to ring the bell to call up Lurch was an arduous task that added a very cumbersome element to having Lurch assist in completing specific tasks. Of course this was done to be funny. Yet in a similar way, although the intent is not intended to induce laughter, calling up a voice assistant is a surprisingly arduous task today. Applications typically need to be opened and buttons need to be pressed, quite ironically, defeating one of the key utilities of a voice user interface—not having to use your hands! For example, if I'm driving and I want to use my phone, I need to reach over and grab it, hit the "on" button, type in my 4 digit password, scroll to the right voice application open it up then hit a microphone button! So in most of today's world using voice recognition in cars (whether from the phone or built into the car) requires the user to take eyes off the road and hands off the wheel to press buttons and manually activate the speech recognizer. Definitely more dangerous, and in many locales it is illegal!

Of course, given the fact that human-factors specialists and speech recognition companies are working overtime to make voice assistants better, such encumbrances will soon fade into the past. I envision a world emerging where the voice assistant grows from being The Addam's Family "Lurch" to "Radar" from the 1970s hit show MASH. Corporal "Radar" O'Reilly was an assistant to Colonel Sherman Potter. He'd follow Potter around constantly; whenever Potter wanted anything Radar was there with whatever Potter wanted, sometimes even before Potter asked for it. Radar could do anything from problem solving to playing drums. Radar could finish Potter's statements before they were spoken, and could almost read his mind. Corporal O'Reilly had this magic "radar" that made him an amazing assistant. He was always around and always ready to respond.

[3] http://en.wikipedia.org/wiki/Lurch_(The_Addams_Family).

From T.V. Characters to Real-Life Mobile Voice Assistants

One may opine that the voice assistants of the future could end up having versions much akin to Radar O'Reilly. And in so doing, they will adapt to each user individually learning their user's mannerisms, habits, and preferences. They will know who is talking by the sound of the voice (speaker identification), and sometimes, as surrealistic as this may sound, they may even sit around "eavesdropping" on conversations occasionally offering helpful ideas or displaying offers to users even before they are queried for help. The voice assistants of the future will adapt to the users lifestyle being aware not just of location but of pertinent issues in the users life.

Let me give you an example that will bring this futuristic technology into the realm of realism. As a vegetarian smartphone user, I have done a number of searches for vegetarian restaurants. Thus, my mobile voice assistant can easily build a profile of me that includes the fact that I like to eat vegetarian dinners when I'm traveling. Drawing on a personal data profile, my virtual assistant may suggest to me, a good place to eat when I'm on the road. The suggestion would probably come at the right and would be somewhere nearby. It would easily know when I'm on the road; consequently it could also figure out by my location whether I had sat down to eat. It probably knows that I don't mind walking as long as it's not too hot out. This future virtual assistant might occasionally show me advertisements but since it will know my habits and preference, the ads will be so highly targeted to my needs at the time the show up that I'd really enjoy hearing about them as opposed to being annoyed by senseless ads. In a similar way, the virtual assistant would function as "Radar" did when he sometimes made suggestions to General Potter to help him in his daily life and challenges!

Combining Embedded Speech Recognition with Cloud-Based Speech Technology

Most of the heavy lifting for VUI's and Voice Search is done in "The Cloud." As such, this is where most of the investment from the big OS (operating system) players has gone. The Cloud offers an environment with virtually unlimited MIPS and memory. But with the growth of cloud based speech technology usage, we have seen bandwidth and speed limitations arise. This bandwidth limitation has led to a lot of complaints about the speech recognition. Sensory has been getting more and more requests for an embedded Siri like approach, so there is no bandwith and connection limitation. I suspect that in the Android JellyBean, a clever mix of embedded and server technology was used to improve the response time.

Embedded speech, which is performed on the device itself, can be the only solution for speech control of the device and for spoken input when no remote service is available. This then makes speech a necessary component which adds value to the user experience. In addition, embedded speech also has the ability to consume fewer

resources. What this means is that for battery consumption alone, embedded speech capabilities in consumer devices wear down the batteries much less than connecting and sending data through the Cloud. In addition, the laws in many states which prohibit holding a mobile phone while driving have made it all the more necessary to speech enable mobile phones so that they work when you just talk to them, and you don't need to hit a bunch of button first. Consumers have come to demand improved accessibility to speech in mobile devices. Such demand has spurred growth in speech synthesis and VUIs that can only be fulfilled when speech is embedded in the device itself, which we refer to below as "on the client."

From what we've seen in the speech industry the optimal scenario for Client/Cloud speech usage entails voice activation on the client, with the heavy lifting of deciphering text and meaning performed in the Cloud. Embedded control of the device (placing speech recognition and spoken language understanding on the client itself) also makes sense, because it allows the device to be controlled if the Cloud connection is gone. And in instances where data reside on the device itself, embedded search makes particularly good sense. In such a case, the typical consumer electronic scenario would be to have the device on and listening so that no button presses or touch are even necessary. It could be "always on" or through some detection scheme could come on when needed to reduce power requirements. This paradigm of "no hands or eyes necessary" is particularly useful in car for safety and at home for sheer convenience. We suspect that as more and more data moves to the device, more and more of the speech capabilities will be on device to access that data.

Key Stages of a TrulyHandsfree User Experience

To create a truly "hands-free eyes-free" user experience in consumer electronics, there are a number of technology stages that would need to be addressed (Fig. 2.1):

Stage 1: Voice Activation. This essentially refers to replacing the button press. The recognizer always needs to be on and ready to call "Stage 2," where speech recognition and transcription occur, into operation; but, most important, the recognizer must be able to activate in very high noise situations. Another key criterion for this first stage is a VERY FAST response time. It must be real time, because if the function is to adequately replace a button, then the response time must be the same as a button, which is near instantaneous, and delays of more than a few hundred milliseconds can cause accuracy issues from users speaking to "Stage 2" before the recognizer is listening. Simple command and control functions can be handled at "Stage 1" by embedded speech recognition platforms that are found in the car; alternatively, a more complex "Stage 2" system could be embedded or cloud based.

Stage 2: Speech Recognition and Transcription. The more power hungry and powerful "Stage 2" recognizer translates what is spoken into text. If, however,

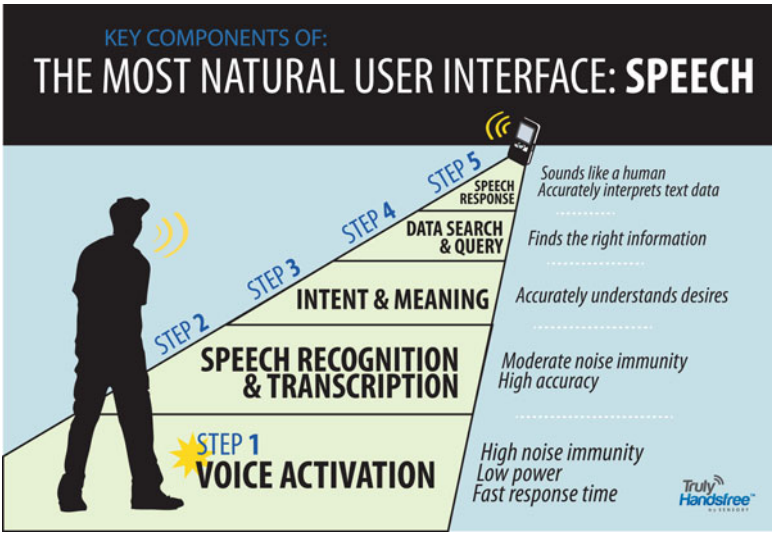

Fig. 2.1 Key components of: the most natural user interface: Speech ©Sensory

the purpose is text messaging (transcription) or voice dialing, then the process can stop here. If the user wants a question answered or cloud based data to be accessed then the system moves on to "Stage 3," which is described below. Because the Stage 1 recognizer can respond in a high noise environment, it can [drop] reduce volume on the [in] car radio or on the home AV to assist in "Stage 2" recognition.

Stage 3: Intent and Meaning. This now gets us to the heart of spoken language understanding and is probably the biggest challenge in the process of the hands free eye free user experience with the voice user interface. The text is accurately translated, but what does it mean? For example, what is the desired query for an internet search? Today's "intelligence" might try to modify the transcription to better fit what it thinks the user wants, because traditionally computers are remarkably bad at figuring out intent of users. And this is exactly why the mobile industry recognizes the urgent need to develop the science of spoken language understanding so that the intent and meaning of the speaker's natural language entries will be better understood by voice driven mobile devices.

Stage 4: Data Search and Query. Searching through data and finding the correct results can be straightforward or complex depending on the query. But we can certainly rest assured that with Google and other online search providers pouring lots of money and time into unscrambling the voice searches of millions of users, performing speech recognition and finding what the user wants will improve rapidly.

Stage 5: Voice Response. A voice response can replace or augment visual displays. Todays' state-of-the-art Text-To-Speech (TTS) systems are highly intelligible and even quite natural sounding.

What Are Some of the Impediments to Voice Activation of Consumer Devices?

There are a number of impediments to the design and use of the fully voice activated consumer device which need to be "always on" in order for voice activation to work effectively. By considering each impediment seriously, speech scientists and system developers can be better equipped with strategies for overcoming such obstacles. Here is a list of the three most common obstacles to voice activation (the first stage), which include performance, response time and power consumption:

1. Accuracy. Buttons often remain the preferred mode of choice for activating the speech recognizer on a consumer device, as opposed to pure voice activation, even though the putative benefits of a "hands-free eye-free" user experience with consumer electronics, especially while driving, are vast. The main reason for this has been that buttons, although distracting and expensive, are quite reliable and responsive, even in noisy environments. Mobile settings constitute challenging environments for speech recognizers: a voice activation word must respond in cars (windows down, radios on, and lots of road noise) and in home (babies screaming, music on, etc.) and without the benefit of the mic just a few inches from the speaker's mouth. Traditional speech technologies function reliably when responding in quiet environments with a close talking mic, but because voice activation in the mobile setting doesn't enjoy such luxury it makes it very hard for the speech recognizer to properly hear the user's voice input needed to activate the mobile device.
2. Response time. The requirement of a speedy response time further complicates this challenge to using voice activation in consumer devices. That is, speech recognizers often require hundreds of milliseconds just to determine the user is done talking before starting to process the speech input. This time delay might be perfectly acceptable during the back and forth dialog between a user and a voice-enabled system, when the speech recognizer is engaged in a litany of question/answer dialog with the consumer. However, at "Stage 1," when voice is used to activate the consumer device, the response of the voice activation is to call up another more sophisticated speech recognizer to naturally proceed to "Stage 2," so that the necessary speech recognition and transcription can be done. The problem is that one cannot expect consumers to accept a delay lasting much more than the time it takes to press a button. Here, we're not only talking about consumer impatience but on a practical/logistic level, the longer the delay, the more likely a speech recognition failure to occur at Stage 2 because users might start talking before the more sophisticated and powerful "Stage 2" recognizer is ready to listen to the user. No doubt the user when speaking before the recognizer is prepared to assimilate the speech input sets off a chain reaction of annoying speech recognition errors.
3. Power consumption. Voice activation requires devices to be always on and listening. Speech activity detectors can be turned on to quickly wake when a person

talks, but even the activity detector requires microphones on and circuitry running. Even devices plugged into walls need "green" certifications and have to prove minimal power consumption when not in use. A number of consumer electronics manufacturers set the target at micro-amps, not milliamps.

Yet, in spite of these obstacles, recent advances in embedded speech technology and battery power conservation achieved through embedded technology in core processors, the use of speech detectors, and low MIPS algorithms have made it possible to have true VUI's without the need to touch devices with one's hands. Voice activation makes it much more convenient for the user's handling of consumer electronics and mobile phones when their hands are busy with other tasks such as driving, operating machinery, in transit, or are covered with industrial gloves or grease, slime or dirt, or when the user just wants the convenience of not having to get up from where he/she is to turn on a device with a button/switch.

Conclusion

The future of speech in smartphones and consumer electronics augurs well. To predict the future let's consider how far we've come in our use of a basic consumer product like the T.V. Decades ago a person watching television had to get up and walk over to the television to turn on the set or switch the channel. Then the arrival of the remote control put an end to all that, and today nobody would consider buying a T.V. without a remote to operate the set. Nevertheless, today we get up and walk over to most of our computing devices to use them by turning them on manually rather than via a remote control as we do with the T.V. But as speech recognition improves, turning on the computer manually will no longer be necessary, and the simple tasks like flipping light switches or controlling HVAC will be done without getting up. Similarly, searching for information or data on one's smartphone might be executed from beginning to end with voice queries and voice feedback. And in keeping with our T.V. characters we discussed earlier, our virtual assistants will be more like "Radar" than "Lurch". Not only will they know you and your habits, desires, wishes, but you, as the user of voice-driven consumer electronics and mobile devices, will shape such assistants to have the personality and characteristics you want!

Sounds impossible? Not so. Today speech scientists are working assiduously to improve spoken language understanding so that mobile phones and consumer devices are capable of engaging in intelligent dialog with the user, even detecting the user's emotions and state of mind. Linguistic subtleties, such as idioms and metaphors, are being analyzed too so that no part of human language is left unexplored by linguists and speech engineers. The end result will be the creation of mobile voice assistants and consumer electronics that are able to respond to spoken language entries with the same ease as they respond to a keystroke or a flick

of a switch. And perhaps then, the decades of hard work of system designers, linguists and engineers, laboring to make computers truly understand natural (human) language, will be better understood by the everyday consumer whose mundane tasks will be completed with much more ease, and perhaps with some delight too!

Chapter 3
The Personal-Assistant Model: Unifying the Technology Experience

William Meisel

Abstract Innovations in the interface between computers and humans have driven rapid expansion in adoption of computer technology and access to the resources of the Web—with the graphical user interface serving as a prime example. Speech and natural language technology has now reached a critical threshold in its development that allows further enhancement of that human-machine connection, driven in part by the limitations of the graphical user interface and clumsy text entry on smaller mobile devices. The "personal assistant" model integrates a number of natural interface technologies, making the increasing number of features and applications available a more unified experience. This chapter defines the Personal-Assistant Model of user interface design, the technologies required to support it, and the directions in which it is likely to evolve.

Introduction

The popularity of Apple's Siri has demonstrated the appeal and practicality of a voice-enabled personal assistant (PA) application. And yet, while the speech recognition is the most visible feature of Siri, it is the interpretation of what is said in "natural language" (Natural Language Processing, NLP) that is critical to making Siri effective. The initial success of some of today's versions of PAs suggests that speech and NLP technology have passed the threshold of usability for a PA application, in keeping with what author Malcolm Gladwell would call the "tipping point." One can expect as these technologies continue to evolve, the personal assistant

W. Meisel, Ph.D. (✉)
TMA Associates, PO Box 570308, Tarzana,
CA 91357-0308, USA
e-mail: wmeisel@tmaa.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_3,
© Springer Science+Business Media New York 2013

approach to interacting with a user will become even more effective over time. This raises some important questions about PAs:

1. What is the best model for personal assistants and how will it evolve?
2. Will technologies beyond speech recognition, text-to-speech synthesis, and NLP be required for personal assistants to reach their full potential?
3. The Graphical User Interface (GUI)—with windows, icons, menus, and a pointing device—drove the popular acceptance of PCs, and is largely the foundation of user interaction with today's smartphones and pad computers. Can a "Personal-Assistant Model" become a dominant user interface method comparable in its impact to the GUI? Can it expand beyond mobile devices to TVs and personal computers?

The Personal-Assistant Model (PAM)

Based on the example of Siri, one might informally define a personal assistant application as software that allows a request spoken by the user in natural language to be fulfilled by the software with sufficient accuracy that the user continues to use the personal assistant. More formally, a personal assistant (PA), as defined in this chapter, is an application based on a Personal Assistant-Model of the user interface:

> A Personal-Assistant Model (PAM) is software that can take a communication posed in a natural language as speech (as a full sentence or in an abbreviated form), interpret the desired intent of that communication, and provide the user with the desired result as directly and accurately as possible, with one option being a voice response. To the degree that there is context or other input or output modalities available that can help get or deliver the desired result (including entering the inquiry as text and delivering a viewable result), the personal assistant may use those resources.

We add a corollary that will be discussed in more detail later in the chapter:

> One personal assistant application may incorporate support from another personal assistant application through an appropriate transfer of control.

The mass media and most users see speech recognition as the most obvious aspect of mobile personal assistants—it's the technology that avoids the difficulty of entering text or navigating by repeated touches on a small screen. It is also the preferred input method because it allows a full NLP request to be entered quickly. As noted, speech input is particularly important on mobile devices (since typing is inconvenient and sometimes unsafe on a small device) or on technology used at a distance, such as a "smart" TV.

However, an option to input an inquiry as text would make the PA available to the user when speech isn't practical for whatever reason (e.g., a noisy environment, privacy concerns, or it would disturb others nearby). Hence, a text input option is necessary if the PAM is to be the primary user interface model in the user's mind.

The utility of a well-designed personal assistant isn't completely dependent on speech. The NLP feature is crucial for getting the user the desired result quickly;

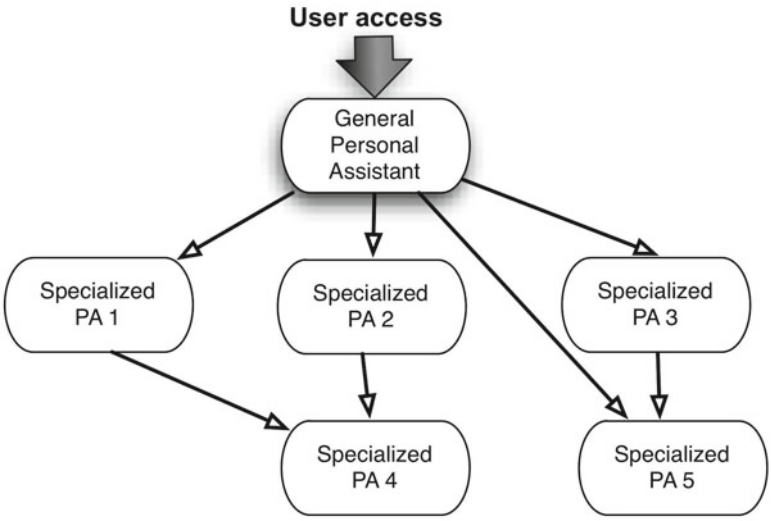

Fig. 3.1 A possible relationship between a managing general personal assistant and specialized personal assistants (PAs)

for example, a typed request for a "Chinese restaurant nearby" can produce both a map and reviews of the two closest Chinese restaurants, with the option to click on one of these results to place a reservation. The personal assistant could be extended to modes beyond speech and text, e.g., gesture recognition or face recognition. If the user has selected a text phrase on a screen, for example, and says, "Is there more information on this?," it is not beyond the scope of a PAM to use that phrase without requiring the text information be repeated. While the PAM, as it defined here, doesn't require that all input to the PA be text or voice, it does require that natural-language text or voice constitute one available kind of input. The output of a PAM is obviously not confined to voice, since a graphical display of results, e.g., a list of web sites, is a typical result of an inquiry with early implementations.

Cooperating PAs

PAs can work within broad or narrow contexts. A general PA, such as Apple's Siri, attempts to deal with almost anything the user requests, defaulting to a reply such as "I can't help you with that" when it can't handle a request. A specialized PA may only deal with a restricted context, such as customer service for a specific company or a competitive product pricing service. As the technology matures, it is likely every company will need a company-branded PA as much as they need a company web site.

PAs can cooperate to offer more than one alone can offer, with specialized PAs acting much like subroutines in software programming. The integration of PAs can be hierarchical, as in Fig. 3.1.

To extend the web site analogy, there may be a number of general PAs, and they may have access to the same specialized assistants, as much as several web sites can reference the same web site. If a specialized PA has the same "voice" as the general PA, the transition to a specialized PA can seem seamless to the user. When the voice/personality of the specialized PA is different than that of the general PA, the general PA can announce the transition: "Bookstore Bob can help you with that. When you want to talk to me again, just say 'Hey, Boss!'" The protocols for such transitions will evolve over time, with the key objective being a degree of consistency so that the user doesn't get lost.

Beyond Smartphones

Smartphones provide the most persuasive case for a PAM: (1) the small screens make the usual graphical user interface conventions and typing harder to use; and (2) loading the many screens that are often required in typical web operations or application use can be slow when wireless data download rates are slow and expensive with wireless plans that charge for the amount of data downloaded. Thus, it is likely that early PAs will be focused on improving the smartphone experience because of these factors.

However, once one has a PA that has learned one's preferences, which has access to personal data such as a contact file, and which has adapted to one's speech and language use, one will most likely want that same assistant available on whatever portable device, PC, or even "smart TV" one is using at the time. The power of the PA to get the user a result quickly may result in demand for the PA to be available everywhere, connected to multiple devices through the Internet. Further, we may want the general personal assistant to be the same in multiple environments so that it can use what it learns about us without having to learn it more than once. A multiplatform PA can also handle a request on a smartphone that impacts another device, e.g., "record American Idol tonight."

This chapter will look first at the technical implications of the PAM, but the role of the PAM is more than technical. As suggested, it has the potential to be the "face" of many different operating systems, unifying much of the growing diversity of approaches to the user interface on mobile and other platforms. I will also examine the importance of unifying the user experience and how the PA can be a key part of that objective.

The Technology Requirements

The availability of a screen and pointing method can improve the utility of a PA. For example, the assistant can display a long list of options when there is no clear single result (e.g., music on the device by a particular artist or a list of applicable web sites)

and allow selection by pointing to the selection. As noted, the PAM includes integrating other input and output modes when those make the most sense. The core technologies that make a PA unique, however, are speech recognition and NLP. Speech recognition and NLP can be synergistic (combined to be "speech understanding"). For example, the NLP need not just interpret the best guess of the speech recognition processing (the one displayed by Apple's Siri, for example), it can utilize "N-best" or another probabilistic representation to see if other close interpretations of the speech "make more sense." However, for the sake of segmenting the discussion, we will discuss NLP as if it is operating on text that has been converted from speech or by direct user text entry.

Natural Language Processing

A core functionality of a PA is dealing with a natural language request *in context*. We are not talking here about some abstract "artificial intelligence" that understands everything, including requests it can't handle. One might get a humorous response to "It's too hot for a walk today," but one doesn't expect a mobile phone to fully understand such statements or act upon them. In fact, with a request such as "What is the current price for Apple stock?", we are asking for it to understand sources available to it on the Web and on the device, sources that in general we might not even know about—we are asking for "computer intelligence."

For a PAM, the NLP can be conceived as having two context-dependent steps in providing this intelligence:

1. *Understanding the general target of the request.* For example, a request "text Joe, I'm on my way" must understand that this is a request to use the device's text messaging functionality. On the other hand, "Find steak houses in Hollywood" is a request that would use search technology and perhaps map services and reviews of restaurants to help select and locate a restaurant matching the request. Understanding the context of the request may involve some memory of what the user has recently done or of the user's preferences—in other words, the broader context.
2. *Finding the variables (the data) necessary to specify and retrieve the desired information:* Once the NLP understands the resources available to it to satisfy the request, it will ideally extract the information from the natural language request that lets it enter the data that the identified applications or data sources need in order to provide the answer to the specific request as directly as possible. For "text Joe, I'm on my way," the NLP should extract Joe as a name to be submitted to the device's contact list to get the phone number for text messaging and "I'm on my way" as the message to be entered in the text message field.

The NLP should find the target of the application and the variables the application needs to complete the request, but custom software is also typically required to complete the request. For example, with the NLP having detected the request is a

text message, custom software must use available Application Programming Interfaces (APIs) to launch the text messaging application and enter the addressee and the text. Similarly, interaction with a calendar/reminder application is typically custom-coded and specific to the application, most often the application that is part of the device's software upon purchase. When going to a particular web site for information, e.g., to Yelp for a review or Wolfram|Alpha for a fact, there is a similar requirement to understand that particular web site and how to enter information from a natural language request into that web site's forms to provide a direct result. Again, this will generally require custom programming specific to each web site, as well as an interpretation of the natural language request as something that the web site can address. The following section suggests that this lack of generality in representing Computer Intelligence is a technical challenge for the PAM that can and should be addressed.

Knowledge Representation

Some of the examples in the previous section suggest the challenge in getting the desired answer to a request. I have pointed out that satisfying a request might involve custom programming to, for example, interface with a calendar application or a specific web site. But with the huge size of the Web and the proliferation of information and applications, customization is only feasible for the most frequently used web sites, sources of information, and applications. Ideally, one would like a technology that can make information available to PAs in a way that minimizes or avoids the need for custom programming, much in the way information on web sites is presented in a standard language that allows its presentation within any web browser that handles HTML. The technology area that takes unstructured text and organizes it has sometimes been called "knowledge representation," and we will adopt that term in this chapter. Knowledge representation uses NLP, but goes beyond NLP, including aspects of what is often called Artificial Intelligence (although, as the reader may have surmised, I prefer the term Computer Intelligence).

The objective of knowledge representation is to take large unstructured sources of information and find the parts of that information that are related and can address specific topics. The technology ideally allows inquiries posed with a list of specific keywords to find the information in "big data," a term that has lately become popular to describe multiple large databases of information that would otherwise require laborious search by a human to find relevant data and correlate it. This data can be, for example, published medical research found in the databases of medical journals and conferences. For a business, it can be company-specific information, data distributed among a company's web sites, user manuals, product descriptions, Frequently Asked Questions documents, instructions for call center agents, and much more.

An Example of Knowledge Representation

Let's look at a simple example that will allow us to see the connection between the use of NLP to determine an objective and the use of knowledge representation to help provide a more specific answer. Suppose a PA is designed to direct a user to the appropriate web page on a company web site. The usual search technology takes the words in the search box on a web site and returns a list of web pages on that site that match the search, perhaps doing a simple frequency match to words on a web page. It then becomes a laborious ask for the user, who has to look at each page to see if it contains the desired information.

In contrast, a PA using NLP may have more information to work with to determine the intent of the search. First, the phrase may be longer because of the option of speech input rather than typing. For example, someone searching on a web site selling books for a book that they heard about might request of a PA, "Show me digital books on the history of computers by professors" instead of typing "history computers" in a search box. Entering those two words in a search box would provide a list of options not restricted to books available in digital form and written by professors. On the other hand, the PA's NLP should understand that "by professors" meant authors that were professors and "digital books" meant a specific book format, but could only use that information if the web site had been analyzed to provide that information in an accessible form.

If knowledge representation were applied to the book web site, it could characterize every target (every web page, even dynamically generated web pages) by a set of characteristics extracted by intelligent understanding of the material being presented on that web site. In the book-selling example, all the pages could have been previously examined and processed to be represented by a summary available to the PA. The summary would be divided by general categories (book title, author names, book description, author description, versions available, price, customer rankings/comments, etc.). Each category would have associated keywords and phrases (e.g., author description might list "professor" as a keyword). The list of keywords could have an importance ranking based on frequency count and/or where they were used (e.g., in the abstract or title rather than a longer description). NLP could optionally be used to list alternative ways of saying the keywords and phrases ("semantic processing"). Pages generated dynamically from a database (likely the case in the book site example) will already have a ready categorization of much of this information in the database.

The PA would be designed knowing the structure that the knowledge representation could deliver, part of the power of a specialized PA. (The obvious specialization in this example is providing access to books on a web site that sells books.) The task of the NLP in the PA would be to first decide which categories (author description, book description, and versions available) that the inquiry was specifying. This task is made more viable because of the small number of categories relevant to the web site being accessed, defined by knowledge representation of the web site. The second task is to associate keywords with each category. In our example, "computer"

and “history” are among the keywords for book description, “professor” for author description, and “digital” for versions available. This would provide a more targeted result than if the terms “history,” “computer,” “professor,” and “digital” were entered in a conventional search box. I tried entering these four words in a search box on a well-known book-selling web site: The top item returned was *Multiliteracies for a Digital Age (Studies in Writing & Rhetoric)*, which is not available in digital form, and there were no books on computer history in the rest of the list returned by the search. This example suggests that knowledge representation could also deliver better results for conventional search engines, perhaps further motivating web sites (and applications) to invest in knowledge representation.

Technology for Knowledge Representation

One example of a complex technology that could be considered including a knowledge representation method is IBM’s Watson technology, which famously defeated two expert contestants on the *Jeopardy!* television show, and is available today to enterprises to extract information from the excess of today’s data sources. For example, using the same technology as IBM Watson, IBM’s Cognos performance management and business intelligence software can take vast quantities and different types of data, and transform them into “actionable insights,” according to company announcements.

In a specific case, WellPoint, the nation’s largest insurer by membership, is using Watson to diagnose medical illnesses and to recommend treatment options for patients in a new system that will be tested at several cancer cancers in 2012. Watson will be able to analyze one million books (roughly 200 million pages of information) and provide responses in less than 3 s, according to published reports.

Memorial Sloan-Kettering Cancer Center (MSKCC) and IBM scientists announced in March 2012 that they were working together to create a decision support tool for doctors that provides an “outcome and evidence-based decision support system” with quick access to comprehensive (but summarized) up-to-date information about cancer data and practices. Nuance Communications, which has a licensing agreement for Watson technology with IBM, is offering a service called Prodigy that will provide similar analysis of a specific company’s data relevant to customer service or other areas.

More accessible technology that serves particular segments of knowledge representation is also available. For example, there is analytics software that summarizes trends in customer service interactions, including speech recognition to understand what callers and agents are saying in voice interactions. A PAM for such applications might be able to quickly answer questions by management such as, “What are recent trends in customer complaints?” While specialized software is typically available to help analyze information such as a database of contact center calls or text chats, it can be difficult to find specific answers quickly in software designed to provide an overview of a large amount of data. A PAM knowledge representation component can be a key part of the utility of such software.

Natural language processing is of course a key part of knowledge representation systems. After all, we are talking about understanding text and speech databases that contain full-text documents such as patents and published research in areas such as healthcare. Knowledge representation must condense that understanding into compartments—summarizing it sufficiently so that one can get to specific answers or at least identify the most relevant knowledge sources. Whether one includes that summarization under the heading of "natural language processing" or treats it as additional functionality of knowledge representation is largely a question of definition. Any particular NLP methodology, including knowledge representation, must have limited objectives to be successful, at least for the foreseeable future. It is dangerous to consider the term analogous to the broad scope of human natural language understanding (which in part depends on decades of experience in a human body).

Ideally, there will eventually be an industry standard analogous to HTML on how web pages or services and software applications present their contents to a PA. In that case, the responsibility of representing the knowledge in the source will be distributed across providers of those web sites and applications.

Usability Improvements and Transferability by Unifying the User Experience

To understand the potential impact of the PAM, let's step back a bit and review an earlier user interface innovation. Xerox's Palo Alto Research Center (PARC, now a Xerox subsidiary that simply goes by the initials) invented the basic Graphical User Interface (GUI) that is still the basis of operating systems on PCs, smartphones, and tablets today, including the Microsoft Windows and Macintosh operating systems. The basic GUI has sometimes been called the WIMP interface—Windows, Icons, Menus (pull-down menus), and Pointing device (initially the mouse). Our familiarity with these intuitive features of devices on PCs (both the Microsoft Windows and Macintosh operating systems) and in Web browsers has made use of their basic features to a large degree intuitive. The extension of the WIMP interface to touch screens (with the pointing device being a finger) is a key part of the user interface on those devices.

This transferability of learned user-interface functionality has driven the rapid expansion of consumer use of digital technology. If large parts of the user interface become exclusive to individual companies and devices, the transferability of a user interface innovation and its advantages for consumers is crippled. Unfortunately, the rush to the patent office today has led to companies trying to make certain user interface innovations proprietary, many of which have now become the subject of patent lawsuits today. If patent suits and other fights over intellectual property force every device to use a different looking and/or acting user interface, moving from one device or supplier to another becomes more of a challenge. If operating systems for PCs, smartphones, and pad computers couldn't use the familiar WIMP GUI,

I suspect the markets for the devices would not have developed as explosively as they have.

Fortunately, language evolved before the age of patents. Certainly, there are and will be patents on methods of NLP but much of the core literature is academic and has been published, making it difficult to legitimately patent anything truly fundamental. And enforcing a patent on an invention that isn't visible but embedded in software would be difficult—imagine trying to explain to a jury a segment of computer code in a competitor's software that a company claims is a patented invention for understanding some aspect of NLP. In any case, it is easier to change the way a particular case in NLP hidden in software is handled to avoid a patent claim than to change a part of the GUI that a user can see.

A PA application as defined in this chapter could appear consistent across platforms and vendors, even if a variety of methods are used to accomplish the speech recognition, NLP, and knowledge representation. The natural language accepted by the PA has been familiar to each of us since not long after birth—no user manual required. The power of a PAM to allow us to move from one device or vendor to another and still have the same user experience—to meet the objective of transferability—has some of the power of the basic GUI to advance the usability of technology further.

And a consistent natural language experience to a large degree reduces the burden on the GUI to handle more and more complexity as applications and web sites expand. One can argue persuasively that the popularity of Apple's Siri is due to its overcoming some of the increasing complexity of using the GUI (e.g., finding a particular feature in an application that has long menus and submenus). Most major published criticisms about personal assistant software are not about the *way* they operate, but about *how well* they operate—a problem that will presumably diminish as the underlying speech recognition and NLP technology improves and if knowledge representation provides more shortcuts to the requested result.

That usability improvement isn't confined to mobile devices. Most of us have experienced the over-burdening of the GUI on PCs as features exploded. With programming on cable TV running to hundreds of channels and "Smart TVs" connecting us to video sources on the Web, the same explosion of options has overwhelmed the current remote-control approach to selecting TV programs and will do the same for a more complex GUI approach. The PA model, if fully developed, can directly answer a question like "why is my word processing program ignoring my instructions to add space before and after a particular paragraph?" (Because the option was checked to not use the spacing if two subsequent paragraphs are labeled with the same style.) It can allow commanding one's TV to "record the Lakers game today and extend the recording one hour beyond the scheduled end."

To be clear, this argument doesn't claim the GUI is obsolete, just overburdened. Synergy is important: GUI and PAM user interface models can help each other when both are available.

PA technology will advance, based on advances in speech recognition, NLP, and knowledge representation, as well as improvements in design and software implementation to use context and other application-specific features. Development of specialized PAs that aid a general PA makes the general PA more powerful.

The integration of PAs is likely to be accelerated by market developments in the rapidly evolving advertising models that have given us many free services. On mobile devices, the display ads popular on web pages get lost. The move to go directly to content competes with displaying a list of web sites headed by advertisers paying to be at the top of the listing. Instead, advertisers may pay to be the preferred answer to appropriate inquiries. Web services such as restaurant review sites may pay to be the site used for such inquiries (and already are in some cases). Marketing payments each time the general PA invokes a company-specific PA make another source of ad revenue. Financial incentives are likely to drive the PAM model to grow more quickly than one might assume.

Will a single general PA be available on multiple devices? Probably. For example, the new Smart TVs are really PCs with Internet connections that we view at a distance. We may want to ask our PA a question unrelated to TV entertainment while comfortably seated on our couch, and thus want it to have the same functionality and personalization it exhibited on a mobile device.

Concluding Summary

Technology advances and growing availability and generality for automated personal assistants should pass a threshold at some point where a user comes to think of the PA as the primary user interface, whether one speaks to it or types a request. The proliferation beyond mobile devices would be encouraged by the PA delivering more precise results directly rather than requiring further searching by the user.

A Personal-Assistant Model is defined in part by its being able to handle a natural language request. A spoken request is the emphasis of currently available mobile personal assistants, but the natural language understanding would add value even if the inquiry to the PA was entered as text. A key feature that drives the utility of a PA is delivering the requested result more directly than a current Graphical User Interface does—with less searching and navigating. To the degree this direct-to-content aspect of a PA is effectively implemented, it can significantly improve the user's experience and make existing applications and services easier to use. Direct-to-content capability can support powerful marketing models that will drive its adoption. Full implementation of the direct-to-content feature may require knowledge representation technology to achieve its full potential; a source of information or services can be made more accessible to a PA through knowledge representation. An industry standard on how knowledge in a source is represented would accelerate the adoption and utility of a PA. The intuitive nature of interacting with a PA will grow over time as the technology improves and the information, applications, and services it addresses become more tightly integrated with the PA. Specialized PAs will act on their own and can also be used in effect as subroutines by a general PA. In the long term, the Personal-Assistant Model has the potential to be the primary user interface modality on many platforms.

Part II
Innovations in Natural Language Processing

Chapter 4
Natural Language Processing: Past, Present and Future

Deborah A. Dahl

Abstract This chapter provides a broad discussion of the history of natural language understanding for both speech and text. It includes a survey of the general approaches that have been and are currently being applied to the goals of extracting the user's meaning from human-language inputs and performing useful tasks based on that analysis. The discussion utilizes examples from a wide variety of applications, including mobile personal assistants, Interactive Voice Response (IVR) applications, and question answering.

Introduction

Enabling a computer to understand everyday human speech or ordinary written language, and do something useful based on that understanding has been a scientific goal at least since Alan Turing proposed that the ability to carry on a believable conversation could serve as a test of a truly intelligent machine in 1950 (Turing 1950). The difficulty of doing this task in its full generality has been consistently underestimated throughout the history of the field. However, in the past 15 years, (and accelerating at an even more rapid pace during the last couple of years) significant progress has been made towards making natural language understanding (NLU) practical and useful.

This progress has not been based on any fundamental, new insights in how human language works. Instead, I would argue, the progress made in NLU is based on factors having to do with the engineering aspects of natural language processing, as opposed to scientific ones. Specifically, (1) recognizing the need for robust processing in the

D.A. Dahl, Ph.D. (✉)
Conversational Technologies, 1820 Gravers Road,
Plymouth Meeting, PA 19462, USA
e-mail: dahl@conversational-technologies.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_4,
© Springer Science+Business Media New York 2013

face of uncertain input; (2) identifying tractable tasks that are less ambitious than full, human-quality NLU; (3) network capabilities that allow systems to leverage the full power of distant servers (4) vast amounts of real data accessible over the Internet; and (5) Moore's Law which allows algorithms that were once impractically resource-intensive to be tested and put into practice.

Today, we have powerful personal assistants, like Apple's Siri, that respond to everyday types of natural language requests like checking the weather, setting up meetings, setting reminders and answering general knowledge questions, very much in the way that early researchers imagined so many years ago. These assistants are far from perfect—Siri makes plenty of mistakes—but they are good enough to be practical, and they are getting better. Let's look at the history of the technology behind these applications to see how this technology has made possible mobile personal-assistants, Interactive Voice Response (IVR) systems and other modern-day uses of NLU.

Beginnings

Natural language processing has been a topic of interest since the earliest days of computing. Early publications such as Claude Shannon's 1948 paper on information theory (Shannon 1948) proposed a statistical theory of communication that considered communication as a statistically-based process involving decoding of a signal; and in fact, some early work was done in the field following this model. However, Noam Chomsky's influential 1957 book *Syntactic Structures* (Chomsky 1957) changed the fundamental direction of natural-language processing research with its claim that the structure of natural language is inherently incapable of being captured by statistical processes. The following 30 years of work followed a path based on formal languages as the primary tool for addressing the problem of NLU. However, in the early 1990s statistics again came to the forefront of NLU. This was at least partly due to breakthroughs in speech-recognition systems, enabled through the use of statistics, such as Lee (1989), as well as the efforts to bring speech recognition and NLU together in programs such as the DARPA Spoken Language Program (1989–1994)[1]

The Process of Natural Language Understanding

All natural-language processing systems take some form of language—whether it's a spoken dialog, a typed input, or a text—and extract its meaning. Some natural-language processing systems go directly from words to meanings while others

[1] See Hirschman (1989) for an introduction to the first workshop.

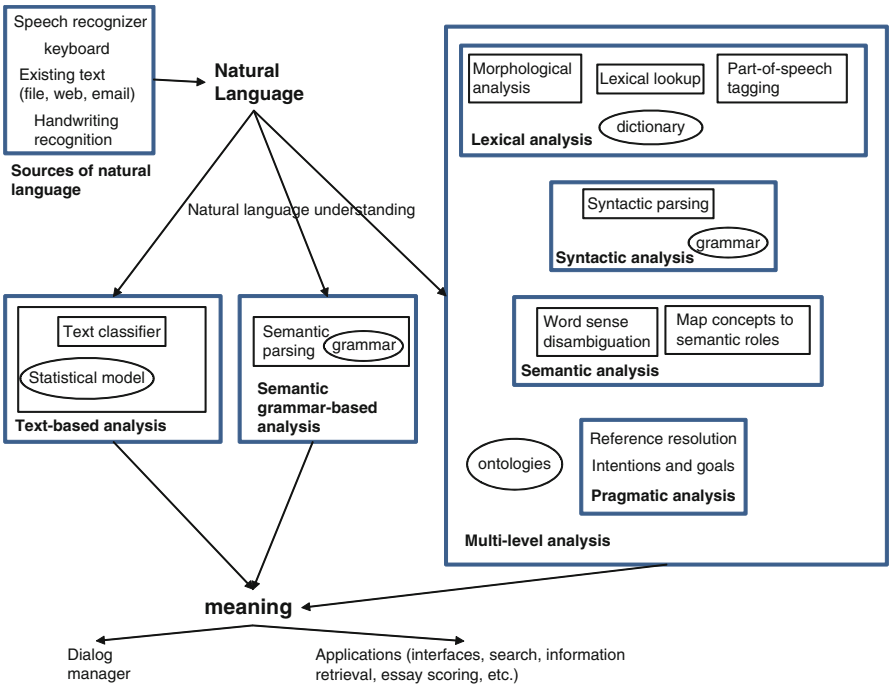

Fig. 4.1 Approaches to NLU

perform one or more levels of intermediate analysis. Systems that go directly from words to meanings are typically used to identify fairly coarse-grained meanings, such as classifying requests into categories or web search. A system that is retrieving the web pages that are relevant to a specific search doesn't need to do a detailed analysis of the query or the documents themselves. On the other hand, finer-grained analysis with more levels of processing is usually used for tasks where the system has to understand exactly what the user has in mind. If I say "I need a flight from Boston to Denver on July 22 that arrives before 10 a.m.," the system needs to extract the exact date, time and cities to provide an answer that satisfies the user.

Figure 4.1 provides a broad perspective on the three main general approaches to NLU.

Natural language can come from many sources as shown in the upper left-hand corner of Fig. 4.1—speech recognition, a keyboard, handwriting recognition, or existing text, such as a file or web page. The goal is to find out what the meaning of that language is, where "meaning" is very broadly understood as some representation of the content of the language that is relevant to a particular application. Figure 4.1 shows three approaches to NLU, labeled *text-based analysis*, *semantic grammar-based analysis*, and *multi-level analysis*, which we will explore in detail in this chapter.

In text-based analysis, the basic unit of analysis is the text itself. Statistical models based on information such as the proximity of different words to each other in the text, the relative frequency of the words in that text and other texts, and how often the words co-occur in other texts are used to perform such tasks as web search and document classification.

In contrast, the other two approaches, semantic grammars and multi-level analyses, both attempt to define some kind of structure or organization of the text to pull out specific information that is of interest to an application. Semantic grammars look directly for a structure that can be used by an application; whereas multi-level analysis looks for multiple levels of intermediate structure that eventually result in a representation of the meaning of the text in a form that is useful to an application. (It should be noted that these are idealized systems; most actual systems contain elements of different approaches.)

As shown in the bottom of Fig. 4.1, after the meaning is produced it can be used by other software, such as a dialog manager or another application. The rest of this chapter will discuss these components in detail, and will conclude with a discussion of integrating natural language processing with other technologies.

Multi-level Analysis

We'll start by looking at the multi-level analysis approach.

Natural-language processing systems which do a detailed analysis of their inputs traditionally have components that are based on subfields of linguistics such as the lexicon, syntax, semantics, and pragmatics. The relative importance of these components in processing the language often depends on the language. For example, analyzing written text in languages that don't have spaces between words, such as Chinese or Japanese, often includes an extra process for detecting word boundaries. Processing can be done sequentially or in parallel, depending on the architecture of the system. Many implemented systems also include some aspect of probability. That is, how to analyze an input may be uncertain when the input is analyzed, but if one of the analyses is more likely, the less likely analyses can be either eliminated or explored at a lower priority. For example, "bank" in the sense of a financial institution is a more likely meaning in most contexts than the verb "bank" in the sense of piling up a substance against something else.

Lexical Lookup

Starting from either a written input or the output of a speech recognizer, lexical lookup describes information about a word in the input. It may include a step of *morphological analysis* where words are taken apart into their components. For example, the English word "books" can be analyzed as "book"+"plural." This is especially important for languages where words have many forms depending on

Table 4.1 Parts of speech for "like"

Example of usage	Part of speech
I like that	Verb
Her likes and dislikes are a mystery	Noun
He was, like, eight feet tall	Interjection
People like that drive me crazy	Preposition
They are of like minds about that	Adjective
Your food is cooked like you wanted	Conjunction

their use in a sentence (highly inflected languages). Spanish, for example, has more different word forms than English, and a word like "hablaremos" would be analyzed as "speak" + "future" + "first person" + "plural" or "we will speak." There are many other languages that are much more complicated than Spanish, and it would be very impractical to list each possible word in a dictionary for these languages. So these words need to be broken into their components.

Related to morphological analysis is a process called *part of speech tagging*, which identifies a word as a noun, verb, adjective, or other part of speech (see Brill, 1992 for an example). This process provides extremely useful information, especially for words that can be used in many different contexts.

The English word "like" is a good example of a word that can occur as at least six different parts of speech, as shown in Table 4.1.

Automatically identifying the part of speech of a word is helpful for later stages of processing, such as parsing and word-sense disambiguation, which we will discuss below, because it eliminates some analysis options. If the system knows that "like" is a verb in a particular sentence, then it can rule out any other possible analysis that uses "like" as a noun.

Parsing

Parsing is a stage in natural language processing which breaks down a sentence into its components and shows how they're related to each other. Parsing can have the goal of finding either syntactic or semantic relationships within an utterance. Syntactic parsing is the older approach, and has been explored in a large body of research since early papers such as Yngve (1960), Marcus (1980), and Woods (1970).

Syntactic Parsing

Syntactic components include parts of speech and phrases, but not the meanings of those words or phrases. Rather, syntactic analysis is based on a set of rules defining the structure of the language. This set of rules is called a *syntactic grammar*. Figure 4.2 shows an example of a simple syntactic grammar that could analyze English sentences like "the cat sleeps on the chair."[2] The first rule states that a sentence

Fig. 4.2 A simple syntactic grammar

Sentence → NP VP
NP → Det (Adj) N (PrepP)
VP → V (NP) (PrepP)
Det → the | a
Adj → blue | white | green | black
PrepP → Prep NP
Prep → under | on | in | behind
N → dog | cat | table | chair
V → sleeps | sits

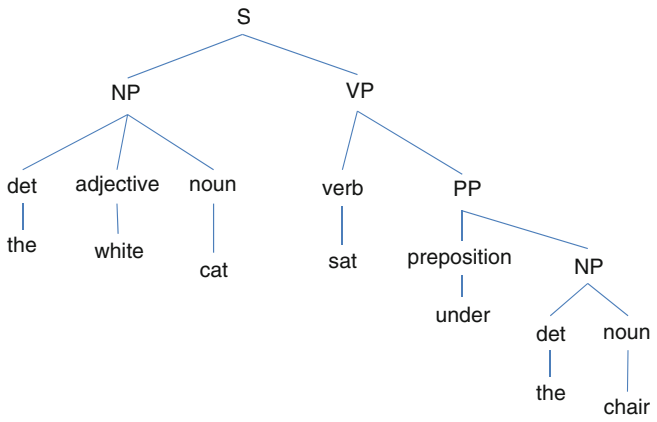

Fig. 4.3 Syntactic analysis for "the white cat sat under the table"

consists of a noun phrase (NP) followed by a verb phrase (VP), and the following rules describe how noun phrases and verb phrases are built, until we get to the actual words (dog, cat, table, etc.) or terminal symbols in the grammar. In this example parenthesized components are optional and alternatives are indicated by "|." Full syntactic grammars for actual human languages are obviously much more complex.

Figure 4.3 shows a syntactic analysis for "the white cat sat under the chair." Because a syntactic analysis doesn't take into account the meanings of the words, we would get the same syntactic analysis for sentences like "the black dog slept

[2]For simplicity, this example is a context free grammar (CFG) in the terminology of formal languages, although normally a syntactic grammar of a natural language would be at least as powerful as a context-sensitive grammar. A commonly used syntax for writing context-free grammars is Backus-Naur Form or Backus Normal Form (BNF), invented by John Backus.

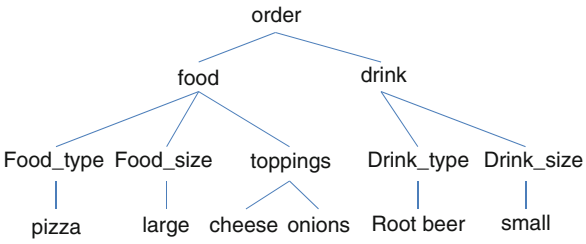

Fig. 4.4 Semantic parse for "I want a large cheese pizza with onions and a small root beer"

behind the sofa," as we would for "the white cat sat under the chair." This is because both sentences have the same syntactic structure, even though they have entirely different meanings. The advantage of an approach that uses syntactic parsing is that the process of syntactic analysis can be decoupled from the meanings of the words that were spoken or even from the domain of the application. Consequently, the syntactic grammar of a language can be reused for many applications. On the other hand, one disadvantage of syntactic analysis is that it is based on a general, domain-independent grammar of a language. Such grammars require a great deal of work to put together even with modern machine learning techniques. However, most applications can be useful without a general grammar. Therefore, other techniques have been developed, most notably parsing approaches that do take into account the semantics of the domain, and whose output is based on semantic relationships among the parts of the utterance.

Semantic Parsing

Semantic parsing analyzes utterances in the context of a specific application such as ordering fast food. Figure 4.4 represents the categories in that application: the type of food, the type of drink, toppings to be put on the pizza, and so on. There is no syntactic information, such as the fact that "pizza" is a noun, or that "large" is an adjective. This is also a less general approach than syntactic parsing in that every new application needs a new grammar. It is, however, generally much faster to develop a one-time, semantic grammar for a single application than to develop a general syntactic grammar for an entire language.

Semantic parsing is also popular for speech systems because the grammar can be used to constrain the recognizer by ruling out unlikely recognition results. Using a grammar to constrain speech recognition supports the fast processing required by the real-time nature of speech recognition. In practice, this means that a grammar used to constrain speech recognizers has to be more computationally tractable than grammars used to analyze text. Speech grammars are always either finite state grammars (FSG) or context free (CFG), as defined in Hopcroft and Ullman (1987).

The result of semantic parsing is a semantic frame, a structured way of representing related information which is popular in artificial intelligence (Minsky 1975). Figure 4.5 shows a complex semantic frame for travel information.

Trip
name: SpeechTEK 2011
departure date: August 7, 2011
return date: August 10, 2011
Transportation to airport
type: taxi
departure: 9:00 a.m.
Flight
airline: United
flight number: 123
departing airport: ORD
departure time: 12:00 p.m.
arriving airport: JFK
arrival time: 3:00 p.m.
Rental Car
company: Hertz
type: economy
pickup time: 4:00 p.m.
Hotel
name: Hilton New York
address: 1335 Avenue of the Americas
city: New York,
state: NY
telephone: 212-586-7000
reservation number: 12345

Fig. 4.5 A semantic frame for travel information

VoiceXML (McGlashan et al. 2004), a widely used language for defining IVR applications, uses semantic frames (or *forms*, in VoiceXML terminology). Figure 4.5 shows a VoiceXML form with *fields* (which correspond to the slots of a semantic frame) which will be filled by the information that the user provides for a card number and expiration date (Fig. 4.6).

Semantic parsing first became popular in the early 1990s as a relatively quick way to get a speech system running. Examples of this approach include Ward (1989), Seneff (1992), and Jackson et al. (1991).

All current commercial grammar-based, speech-recognition systems use semantic parsing.

As speech-recognition systems began to mature during the 1990s, the need for standard ways to write grammars became apparent. Initially, every recognizer had its own format, which made it extremely difficult to use a different recognizer in an existing system. Tools like the Unisys Natural Language Speech Assistant were developed to allow grammars to be authored in a recognizer-independent fashion with a graphical tool that would generate multiple grammars in the various formats. There were also a number of efforts to develop open grammar formats that could be used by multiple recognizers. These included Microsoft's Speech Application Programming Interface (SAPI) grammar format and Sun's Java Speech Grammar Format (JSGF) format. The JSGF format was contributed to the World Wide Web Consortium's (W3C's) Voice Browser Working Group in 2000 and became the basis of the ABNF format of the W3C's Speech Recognition Grammar Format (SRGS)

```
<form>
   <prompt>Welcome to the electronic payment system.</prompt>
   <field name="card_number">
         <prompt> Please enter your credit card number? </prompt>
         <grammar
src="http://www.ajax.com/credit_card_number.grxml"/>
   </field>
   <field  name="date">
         <prompt>Please enter your expiration date </prompt>
         <grammar
src="http://www.ajax.com/credit_card_date.grxml"/>
   </field>
</form>
```

Fig. 4.6 A VoiceXML form with "card_number" and "date" fields

Fig. 4.7 An SRGS rule for a fast food order

```
<rule id="order">
    I would like a
    <ruleref uri="#drink"/>
    and
    <ruleref uri="#pizza"/>
</rule>
```

specification (Hunt and McGlashan 2004), which became a formal standard in 2004. Because Extensible Markup Language (XML) (Bray et al. 2004) was rapidly increasing in popularity at this time, the SRGS specification also defines an XML version of the grammar standard. While the ABNF format is more compact that the XML format, the XML format is much more amenable to machine processing since there are many tools available for editing and validating XML documents.

Figure 4.7 shows an XML SRGS grammar rule for a fast-food order that would enable a recognizer to recognize sentences like "I would like a coke and a pizza with onions." The <ruleref> tags point to other rules that aren't shown here that recognize the different ways of asking for a drink ("#drink") and the different ways of describing a pizza ("#pizza").

The existence of a standard grammar format for speech recognizers made it possible to use grammars to constrain recognition in a vendor-independent way, but that didn't solve the problem of representing the meaning of the utterance. To address that need, SRGS provides for inserting semantic tags into a grammar that would do things, for example, like expressing the fact that whatever was parsed in the "drink" rule should be labeled as a drink. However, SRGS doesn't define a format for the tags. Another W3C standard, Semantic Interpretation for Speech Recognition (SISR) (Van Tichelen and Burke 2007) defines a standard format for semantic tags that can be used within an SRGS grammar. Figure 4.8 shows the rule

Fig. 4.8 SRGS rule with SISR semantic tags

```
<rule id="order">
    I would like a
    <ruleref uri="#drink"/>
    <tag>out.drink = new Object();
       out.drink.liquid=rules.drink.type;
        out.drink.drinksize=rules.drink.drinksize;
    </tag>
    and
    <ruleref uri="#pizza"/>
    <tag>out.pizza=rules.pizza;</tag>
</rule>
```

from Fig. 4.7 with semantic tags. The tags are written in ECMAScript 237 (2001), a standardized version of Javascript. This rule is essentially building a semantic frame that includes "drink" and "pizza" slots. The "drink" slot in turn has slots for the liquid and size of the drink. So, the reference to "out.drink.liquid," for example, means that the "liquid" value of the "drink" frame will be filled by whatever matched the drink in the user's utterance. If the user said "Coke" that value would be "Coke," if the user said "lemonade," the value would be "lemonade," and so on.

The semantic frame that is generated by this rule is the final result of NLU in the semantic-parsing paradigm. It is ready to be acted upon by an application to perform a task such as an interaction with an IVR (e.g. ordering fast food or making travel plans) or a web search. The W3C EMMA (Extensible MultiModal Annotation) specification (Johnston et al. 2009) provides a standard way of representing the output semantic frame as well as other important annotations, such as the time of the utterance and the processor's confidence in the result.

We've brought the utterance through speech recognition and semantic analysis, to a final representation of a meaning that can be used by an application. (How natural language results can be used in an application is something we'll address in a later section.)

At the beginning of the parsing section we described another approach to parsing: syntactic parsing. Looking back at Fig. 4.3, it is clear that a syntactic analysis is not at all ready to be used by an application. So let's return to the syntactic parse in Fig. 4.3 and talk about what other steps need to be taken to finish getting the meaning from the utterance once the syntactic parsing has been accomplished. Once we have a syntactic analysis of the input, the next step is semantic interpretation.

Semantic Analysis and Representation

The process of semantic interpretation provides a representation of the meaning of an utterance. In the semantic-parsing approach discussed above, the processes of looking at the structural relationships among words and deciding the overall meaning of the utterance were not differentiated. This can be efficient, especially for simpler applications, and as we have said, this is the way that all current grammar-based,

speech-recognition applications work. However, it is also possible to separate syntactic analysis from semantic interpretation. This has been done in research contexts, in some earlier commercial systems (Dahl et al. 2000), as well in some very new systems such as IBM's Watson (Moschitti et al. 2011).

Representation

We start with the goal of semantic analysis: obtaining the meaning of an utterance or text. We know what texts and utterances are, but what does a "meaning" look like? We saw one example in Fig. 4.5, a semantic frame with slots and fillers (or attribute/value pairs) like "destination: New York." This is still a very common type of representation. However, many other types of semantic representations have been explored in the past. There have been a number of approaches based on formal logic, for example the research system described in Alshawi and van Eijck (1989) and the commercial system described in Clark and Harrison (2008). In these systems meanings are expressed as logical expressions. For example, "the flight from Philadelphia to Denver has been cancelled" might be expressed as the following

$$\exists(x)\left(\text{flight}(x) \wedge \text{from}(\text{Philadelphia}, x) \wedge \text{to}(\text{Denver}, x) \wedge \text{cancelled}(x)\right)$$

This is read "There is something, "x," which is a flight, and which is from Philadelphia and is to Denver and is cancelled."

Another interesting type of semantic representation is similar to semantic frames, except that the slot names are application independent. These are often called *case frames*. For example, in a sentence like "send an email to Richard" the subject of the verb "send," that is, the understood subject of the command, is classified as an "agent" slot because the subject is acting. The email is classified as a "theme," and the recipient is assigned to the slot "goal." The idea of case frames for semantic representation originated in Fillmore's work (Telephony Voice User Interface Conference) (Fillmore 1985) and was later elaborated in the work of Levin (1993), which presents a detailed analysis of hundreds of English verbs. This approach supports very generic, application-independent systems because the case frames themselves are application independent. Dahl et al. (2000), Norton et al. (1991), and Palmer et al. (1993) are examples of systems that used this approach. On the other hand, the disadvantage of this approach is that, because the slots are application-independent, they still need to be associated with application-specific slots before they can be integrated with an application.

Word-Sense Disambiguation

Another important aspect of semantic processing is word-sense disambiguation (WSD). Many words have more than one meaning, a phenomenon that is called "polysemy." For example, "bill" can refer to something you pay, or a bird's beak. A "tie" can refer to something men wear around their necks or to a game where both

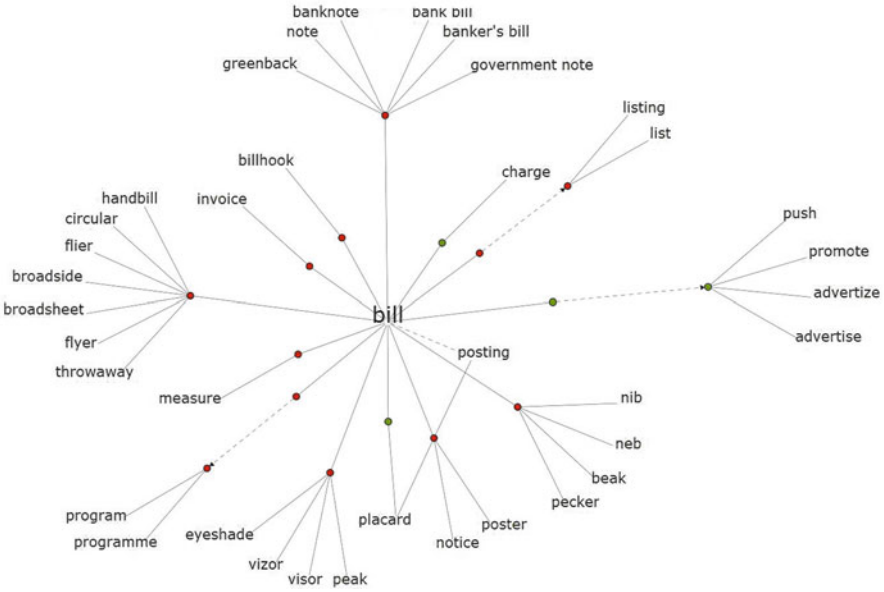

Fig. 4.9 The WordNet senses for "bill"

teams have the same score, and so on. In order to unambiguously represent the meaning of an utterance or text, the correct senses of words need to be assigned. WSD is especially important in machine translation, because words that are polysemous in one language usually must be translated into two different words. For example, the word "hot" in English can refer to temperature or spiciness, but Spanish uses two different words, "caliente" and "picante," for the two concepts. The strategy for WSD is to examine the context around the polysemous words, rule out senses that are impossible in that context, and then select the sense that is most probable in that context from the remaining senses.

Word sense disambiguation often relies on a resource called an *ontology*. An ontology is a structured representation of concepts and their relationships, and defines what the senses are for a particular word. A well-known example of an ontology is WordNet (Fellbaum 1998), developed at Princeton University. WordNet was originally developed for English, but WordNets for a number of other languages have been developed. Figure 4.9 shows the senses in WordNet for the word "bill." As Fig. 4.9 shows, "bill" has six senses in WordNet: a bird's beak, a handbill, the brim of a hat, a banknote, a billhook, or legislation (not counting the proper name "Bill.") A WSD component would have the task of assigning one of those senses to a particular occurrence of "bill" in an input.[3]

[3]The WordNet visualization shown in Fig. 4.9 was created using the Google Code project "Synonym".

Fortunately, the general problem of sense disambiguation can be avoided in many applications. This is because either the topics are not broad enough to include multiple senses, or the number of polysemous words in the application is small enough that the appropriate contextual information can be hand coded. For example, I could ask my mobile personal-assistant a question like "what are the times of my next two appointments" or "what is five times three." The word "times" has two different senses in those two requests, but the senses can be distinguished because they occur in different contexts. In a personal-assistant application, the application is specific enough that the contexts can be hand coded. In this example the developer could simply specify that if the words on either side of the word "times" are numbers, "times" means "multiplication." Work on the more general problem of WSD in broader domains such as newspapers, translations, or broadcast news relies heavily on automated ways of acquiring the necessary contextual information.

To sum up, at the end of the semantic processing phase, we have an exact description of the meaning of the input. It might be represented in any of a number of ways: as an application-specific semantic frame, a logical form, or an application-independent set of case roles, among others, and the senses of polysemous words have been disambiguated.

The next stage of processing is pragmatic processing. As with syntactic analysis and semantic interpretation, not all systems perform pragmatic analysis as a distinct step.

Pragmatic Analysis and Representation

Pragmatics is the subfield of linguistics that deals with the relationship of language to its context. By "context" we mean both the linguistic context (i.e., what has been said before in a dialog or text) as well as the non-linguistic context, which includes the relationship of language to the physical or social world. If I point to something, and say "I like that," the understanding of both "I" and "that" depends on the state of affairs in the world: who is speaking and what they're pointing to. Moreover, use of the present tense of the verb "like" ties the speech to the current time, another aspect of the non-linguistic context.

Reference Resolution

An important and unsolved problem in NLU is a general solution to understanding so-called *referring expressions*. Referring expressions include pronouns such as "I" and "that;" *one*-anaphora, as in "the blue one;" and definite noun phrases, such as "the house." This task is called *reference resolution*. Reference resolution is the task of associating a referring expression ("I," "he," "the blue one" or "the house") with a *referent*, or the thing that's being referred to. Reference resolution is difficult because understanding references can require complex, open-ended knowledge. As in other areas of NLU, the need for a general solution to reference resolution has been finessed in practice by addressing simpler, less general, but nevertheless useful

problems. For example, in an IVR application, the system never really has to interpret "I," even though it is used all the time ("I want to fly to Philadelphia") because there is never more than one human in the conversation at a time. Other pronouns are rarely used in IVR applications. You could imagine something like "I want to fly to Philadelphia. My husband is coming, too, and he needs a vegetarian meal." If the user did say something like that the IVR would need to figure out what "he" means, and that one passenger on this reservation needs a vegetarian meal. However, in practice, speech directed at an IVR is much simpler and consequently the system rarely has to address interpreting pronouns.

Ontologies, as discussed above, also provide useful information for pragmatic analysis because they represent conceptual hierarchies. Some references can be interpreted if we know what kind of thing a word refers to. For example, knowing that "Boston" is a city provides the information needed to know that "the city" in "If I fly into Boston, what's the best way to get into the city?" refers to Boston.

Pragmatic analyses in commercial systems are normally represented in semantic frames where any context-dependent references have been resolved. For example, a user might say "I want to schedule an appointment for tomorrow" instead of a specific date. Because "tomorrow" is a word that must be interpreted with information from the non-linguistic context, pragmatic processing has to identify the actual date that "tomorrow" refers to. The final semantic frame would then include the specific date for the appointment, rather than just the word "tomorrow."

Named Entity Recognition

Another example of tying language to the world is in the task of *named entity recognition*, or identifying references to people, organizations or locations through textual descriptions. Named entity recognition is a type of reference resolution where the referent is an actual individual, place or organization. The descriptions can be extremely diverse, but if an application needs to associate events and activities to an individual, it's important to identify the individual, no matter how the reference is expressed. For example, someone might refer to Barack Obama as "the President" (assuming that we know we're talking about the United States and we're talking about the current president), "the Commander in Chief," "Mr. Obama," "he," or more indirectly, as in "the winner of the 2008 presidential election," or "the author of *Dreams from my Father*." This is a very active research area, and researchers are looking at a number of interesting questions, such as how to recognize named entities in tweets (Liu et al. 2011).

Sentiment Analysis

Sentiment analysis is a new and important application of natural language processing that looks at an aspect other the literal meaning, or *propositional content*, of an utterance or text. All of the types of processing that we've talked about

so far have addressed the goal of extracting the literal meaning from natural language. In contrast, the goal of sentiment analysis is to characterize the speaker or writer's attitude toward the topic of the text. As reviews of products and businesses proliferate on the Web, companies that are interested in monitoring public attitudes about their products and services are increasingly turning to automated techniques such as sentiment analysis to help them identify potential problems. Sentiment analysis tries to classify texts as expressing positive, negative, or neutral sentiments, and can also look at the strength of the expressed sentiment. Sentiment analysis of written texts is technically a type of text classification, which will be discussed in the next section in detail. However, in sentiment analysis, the classification categories have to do with attitudes rather than specific topics. Initial work on sentiment analysis in text is described in Turney (2002). Sentiment analysis can also be done using spoken input, using information such as prosody, which is not available in texts. For example, Crouch and Khosla (2012) describes using prosody to detect sentiments in spoken interactions.

Text Classification

Looking back at Fig. 4.1, we note that we haven't really touched on the text-based approaches to NLU. As we said in the discussion of Fig. 4.1, text-based approaches map inputs fairly directly to meaning, without going through the levels of intermediate analyses that the semantically based or the multi-level approaches perform.

One way to think about text classification is that the goal is to take some text and classify it into one of a set of categories, or bins. Ordinary web search is a kind of text classification. In the case of web search, the bins are just "relevant to my search query" or "not relevant to my search query." The classification result is assigned a score (used internally) so that higher scoring, and presumably more relevant, web pages are seen first by the user. There are many text-classification techniques available, primarily based on machine-learning methods. For example, Naïve Bayes, vector-space classifiers, and support-vector machines are used in text classification, to name only a few. This area is a very active field of research.

Text classification, in combination with statistical speech recognition based on statistical language models (SLM's), has become very popular in the last 10 years as a tool that enables IVR systems to accept more open-ended input than is typically possible with hand-constructed, semantic grammars. Unlike semantic grammar-based systems, the speakers' utterances do not have to match exactly anything that was directly coded during system development. This is because the matching of text to bin is not all or none, but statistical. Combining text classification with statistical-language models of speech was first proposed in Chu-Carroll and Carpenter (1998) and has become very successful. Users are typically much more satisfied with systems that allow them to express themselves in their own words.

As these systems have been deployed in IVR systems and other spoken-dialog systems, a number of refinements in best practices have been learned. For example,

User: "I've been on in and out of the hospital and I know I'm late on it and I'm… I'm… I'm wondering, I'm out of the hospital now and they finally took my cast off, but I still can't work and I can't walk and I'm wondering…."
Classified as "Caller would like to get an extension on paying his utility bill"

Fig. 4.10 Correct processing of an open-ended user request in an IVR

users have more success if the system's opening prompt is not as open ended as "How may I help you?" because users may not understand how to respond to this kind of very open prompt. A more constrained prompt, such as "Please tell me the reason for your call today" is usually more effective.

Because these are statistically-based systems, a drawback to SLM systems is that they require collection of significant numbers of the utterances that are used to train the system, up to tens of thousands in some cases. Moreover, not only must these training utterances be collected, but they must also be manually classified into their appropriate categories by human annotators. This is because the system develops the statistical preferences that it will use to categorize future utterances on the basis of human-annotated data. Training based on human annotation, or *supervised training*, is an expensive procedure. For this reason, training with little or no attention from human annotators, called *unsupervised training*, or *weakly supervised training*, is an important goal of work in this area, although the problem of effective unsupervised training is far from solved.

However, once trained, these systems can be very accurate. The expense of human annotation can be cost-effective in some larger-scale applications, if the alternative is sending the caller to a human agent. Figure 4.10 shows an example of how accurate these systems can be (Dahl 2006), even on very indirect requests.

Commercial systems based on this technology are often referred to as "natural language systems," because they can effectively process users' unconstrained, natural language, inputs. However, as we have seen in this chapter, natural language systems are much more general than this specific technology.

Summary of Approaches

We have reviewed three general approaches to NLU: multi-level approaches semantic parsing approaches and text-based approaches.

1. The multi-level approaches include several levels of linguistically-based analysis, each building on the previous level. These include lexical analysis, syntactic parsing, semantic analysis, and pragmatic analysis. The claim of these systems is that by developing a set of application-independent resources (dictionaries, grammars, semantic information and ontologies), the task of developing new applications

can be greatly simplified. In practice, however, the application-independent resources on which these systems are premised have proven to be extremely expensive and time consuming to develop. Organizations with extensive development capabilities can still create these kinds of systems. For example, the Watson Jeopardy-playing system implemented by IBM is a multi-level system (Moschitti et al. 2011). Unlike the Watson project, most natural-language processing application-development efforts have constrained budgets and cannot afford to develop these resources on their own. In a few cases government funding has enabled the creation of shared resources. Comlex (Common Lexicon) (Grishman et al. 1994) and WordNet (Fellbaum 1998) are notable examples. They are exceptions because in general the required resources are not widely available and must be constructed by each organization.

2. Semantic-grammar based approaches were particularly useful for early speech applications, through the 1990s and early 2000s, because semantic grammars (an example can be seen in Fig. 4.8) are sufficient to process the utterances that were found in limited domains, such as banking or air travel planning. In addition, the semantic grammar serves a useful role in constraining the speech recognizer so that it will only recognize utterances that are appropriate to the application. This significantly improves the accuracy of speech recognition. Semantic grammars are, however, difficult to maintain, especially as the complexity of the application increases. Nevertheless, the vast majority of current IVR applications use this approach. Fortunately, most IVR applications do not require complicated grammars, making this approach highly effective for IVR applications.
3. Text-based approaches became popular in speech applications in the early 2000s, as developers realized that more natural input to IVR's was highly desirable. It was impossible to create semantic grammars broad enough to recognize this more open input, so the text-based approaches came into general use. The large amount of annotated training data that these systems require makes them expensive to build and maintain. This is particularly true if the data changes dynamically, which is the case for seasonal applications. A seasonal retail application, for example, needs new data for each new product added to the application because new products introduce new words for users to say.

Clearly, no single approach is ideal. Each application has its own goals and requirements, making some approaches better for some applications than others. Limited applications like IVR's do well with semantic grammar approaches. Multi-level systems are a good approach for very broad question-answering systems that require a detailed analysis of the questions, like IBM's Watson. Text-based systems are good for classification tasks that require only a general understanding of the input.

Many current systems are hybrids, and incorporate techniques drawn from several of the generic approaches. Mobile personal assistants like Apple's Siri, for example, make use of multiple techniques. Text-processing techniques enable mobile personal-assistants to work with wide-ranging input on unpredictable topics such as web searches from millions of different users. On the other hand, in many cases the inputs to mobile personal-assistants require more detailed understanding

of the user's request. A request that includes a specific date or time needs to be analyzed in detail so that the date or time is handled correctly. A semantic grammar that parses dates and times is the perfect tool for this. Even simple word-spotting can be used, although sometimes that produces incorrect results. For example, a comment to Siri such as "I need $100" gets the response "Ok, I set up your meeting for tomorrow at 1 p.m." Clearly Siri must only be paying attention to the word "one" in that query. Clearly these mobile personal-assistants use an eclectic mix of techniques because of the many different types of conversations they have with their users.

Methodology: Getting Data

All natural language based systems are based on data. In a multi-level system the data may be in the form of dictionaries or syntactic grammars. In a semantics-based system the data may take the form of a semantic grammar. Text-classification systems rely on associations between texts and their classifications (training data) which allow them to classify new texts based on their resemblance to the training texts. Similarly, any kind of system that makes use of probability will derive its probabilities from training data. Early systems used data hand coded by experts, which was time consuming and expensive. As machine learning became more sophisticated, many systems began to use training data that was annotated with the correct analysis by humans without using data that was explicitly hand coded by experts. Human annotators, while expensive, are much less expensive (and more available) than grammar experts. For example, an extensive annotation effort at the University of Pennsylvania, Treebank (Marcus et al. 1993), provided a large set of syntactic parses prepared by humans which were intended to be used in machine learning of parsing techniques. A similar effort, PropBank (Palmer et al. 2005), added semantic case-frames to the Treebank data. Treebank and PropBank represent the supervised approach to annotation. As discussed earlier in the section on text-based approaches, unsupervised approaches require less attention from human annotators but much research needs to be done before unsupervised techniques are good enough for widespread use. At this point, the general problem of data acquisition has not yet been solved.

"Frequently Bought With"

Natural language processing can be part of many other types of systems and often serve as only one component of a complete system. Here we review some of the other components that are often combined with natural language processing. We will focus on interactive dialog systems, like mobile personal-assistants.

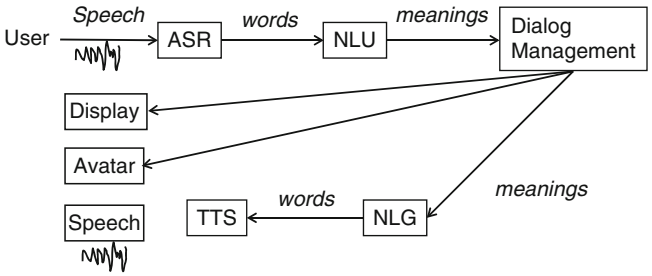

Fig. 4.11 A generic interactive spoken-dialog system

Figure 4.11 shows the complete architecture for a typical spoken-dialog system. As Fig. 4.11 shows, the natural language processing component is only one part of the larger system.

Speech Recognition

Early work on dialog systems with spoken input was part of the DARPA Speech Understanding Program (SUR) of the 1970s (Woods et al. 1972; Erman et al. 1980; Barnett et al. 1980; Wolf and Woods 1980). These were strictly research projects, since the speech recognition of the time was too slow and inaccurate for practical applications.

In the early 1990s speech recognition started to improve dramatically. This improvement was stimulated by two factors. One factor was a technical breakthrough: the use of Hidden Markov Models (Rabiner 1989). The second was the series of formal evaluations of recognition accuracy conducted by NIST (Dahl et al. 1994), which helped researchers understand how each specific algorithmic improvement contributed to overall recognition accuracy. These improvements made possible the development of speech-enabled IVR applications, which continue to be very successful.

At the same time, the formerly favored multi-level approaches (Norton et al. 1992; Austin et al. 1991) were being replaced by the less resource-intensive, semantic-parsing approach (Ward 1989). As discussed earlier, semantic parsing-based methods work best in limited applications, such as checking on banking information. This is because in these systems, every possible input has to be anticipated by the developers. So-called *out of grammar* or *out of vocabulary* utterances cannot be processed. If the user says something that the developer had not anticipated, the system has to engage the user in a tedious dialog to try to get the user to say something it knows how to process.

In order to support more general applications, such as personal assistants, speech recognition has to be able to accept much less constrained inputs. Fortunately,

while parsing-based IVR applications were spreading, the technology needed for recognizing less constrained inputs was being developed independently in the context of speech recognizers used for dictation (e.g., Dragon Dictate and Dragon Naturally Speaking). Dictation systems use Statistical Language Models (SLM's) to define the expected possibilities of words in a user's utterance, rather than grammars. These possibilities are expressed as word pairs (*bigrams*); triples (*trigrams*); or more generally, as *N-grams*. SLM's contain information, such as the fact that the sequence "the cat" is more probable than the sequence "the it." Because this information is probabilistic rather than absolute, recognizers using the SLM approach are more flexible than grammar-based recognizers for recognizing unexpected input.

Dictation technology has continued to improve as it is applied to tasks like web search, which allows for the collection of vast amounts of data from millions of users. The result is now that dictation speech recognition works reasonably well in the context of spoken-dialog systems such as mobile personal-assistants, although factors like noise and accents still affect recognition accuracy.

Multimodal Inputs

Devices that include a display, keyboard, touchscreen and/or mouse enable the user to interact with the device in ways other than voice. This style of computer-human interaction is called *multimodal interaction*. Multimodal interaction has been a research topic for many years (see Bolt 1980; Taylor et al. 1989; Rudnicky and Hauptmann 1992 for early work). However, several factors prevented this early research work from being widely used in commercial systems. One major factor was that speech recognition was not as accurate as needed to support seamless multimodal interaction (error correction was a constant distraction from the user's goals); another was that, for many years, spoken input was limited to a few specific situations. For example, in telephone-based, IVR applications, the alternative is touchtones, which are even more cumbersome than speech.

Another type of application where even error-prone speech recognition made sense was where the user was, for some reason, unable to use a mouse or keyboard. This included users who used speech as an assistive device or users in hands-busy situations. Now, we have both much better speech recognition as well as powerful small devices with keyboards that are difficult to use. The combination of better speech recognition with small keyboards makes spoken and multimodal interaction much more appealing than in the past. In a mobile application like Siri, users can either speak a request, or in some cases, interact with Siri using a touch alternative. The touch alternative is especially useful for tasks such as confirming or canceling a request. AT&T's Speak4it multimodal mobile assistant also supports simultaneous speech and drawing input. For example, Speak4it allows a user to draw a circle on a map while saying "Show me Italian restaurants around here."

In addition to enabling input combining spoken interaction with touchscreens, today's mobile devices routinely include other capabilities that provide additional opportunities for multimodal interaction. These include cameras, accelerometers, and GPS technology. There are also special-purpose sensors that can be added to mobile devices, such as glucose meters or blood pressure meters. These special-purpose sensors provide even more opportunities for multimodal interaction.

Because the number of different modalities continues to increase, it is important to have generic, modality-independent ways of representing inputs from a wide range of modalities. The W3C's Extensible Multimodal Interaction (EMMA) specification (Johnston et al. 2009) provides a way to manage inputs from an open-ended set of modalities. In order to do this, EMMA defines a uniform standard for representing inputs from any modality, whether it is speech, keyboard, touchscreen, accelerometer, camera, or even future modalities. The meaning of the input is represented in the same way, independent of the modality. For example, if the user says "where are some Italian restaurants around here" to her mobile device, the meaning as represented in EMMA would look the same as if the user typed the same request. The modality (speech or keyboard) would be represented as a property of the meaning, but the interpretation itself would be the same.

Dialog Processing

Looking back at Fig. 4.11, we see that the meaning resulting from NLU process can be sent to application components or to a dialog manager. For interactive applications, a dialog manager is very important, since it is the component of the overall system that decides how to act on the user's request. It does so either by reacting to the user with a system response or by taking action to accomplish a task, or both.

The most commonly-used tool for dialog management in commercial systems is VoiceXML (McGlashan et al. 2004; Oshry et al. 2007). As discussed earlier, VoiceXML is an XML language that defines a set of slots (called a "form" in VoiceXML) along with system prompts associated with each slot and speech-recognition grammars that are used to process the user's speech and extract the user's meaning from the utterance. Figure 4.6 shows an example of a VoiceXML form. Originally, the grammar associated with a VoiceXML form was always a semantic grammar in SRGS (Hunt and McGlashan 2004) format; however, with the popularity of statistical natural language processing based on SLM's and text classification, today the URL for a VoiceXML grammar often points to a statistical SLM recognizer.

There is also a considerable research literature on dialog management, particularly task-oriented dialog management (Allen et al. 2000). Major approaches include systems based on planning (Bohus and Rudnicky 2003; Sidner 2004), information states (Larrson and Traum 2000), and agents (Nguyen and Wobcke 2005). (Jokinen and McTear 2010) provides an excellent overview of commercial and academic approaches.

Text to Speech Technology

Spoken output from a system can be provided by audio recordings, as it is in most IVR systems. Synthesized speech can also be used, and is required when it is impossible to pre-record every possible system response. The technology for synthesizing speech from text is called Text to Speech (TTS). There are two general approaches to TTS: *Formant-based* synthesis creates speech from rules describing how to generate the speech sounds; *concatenative synthesis* creates speech by piecing together snippets of prerecorded speech. Concatenative TTS is generally considered to sound better, but formant-based synthesis has a much smaller memory footprint because it doesn't require a large database of prerecorded speech. It is therefore very practical to run formant-based synthesis locally on devices, which is important for minimizing latency.

Application Integration

NLU is not very useful unless it's able to accomplish tasks through interfaces to other software. Certainly, there are applications that just have a conversation, such as the very early program ELIZA (Weizenbaum 1966) or more modern programs called "chatbots," but most practical applications need an interface to other software that actually does something. For a personal-assistant program like Siri, this includes being able to access programs running on the device, like the user's calendar and contacts, as well as being able to access external software such as Wolfram Alpha (Claburn 2009). Siri can also access some device hardware, such as the GPS system. GPS information enables it to answer questions such as "Where am I" (although not similar questions such as "What is my exact location?"). However, Siri cannot access other hardware, such as the camera. Surprisingly little work has been done on the principles of integrating language with external systems; however, Ball et al. (1989) and Norton et al. (1996) describe a rule-based system for integrating natural language processing results with other software and Dahl et al. (2011) discuss an XML interface to external services.

Summary

This chapter has reviewed the history of natural language processing and discussed the most common general approaches: multi-level analysis, semantic approaches, and approaches based on statistical text-classification—using examples from such applications as IVR applications and mobile personal assistants. The chapter also places natural language processing in the context of larger systems for spoken and multimodal dialog interaction. In addition, it reviews related technologies, including speech recognition, dialog management, and text to speech. Today's

NLU applications are extremely impressive, and the pace of their improvement is accelerating. Looking to the future, it is clear that these applications will become even more capable. These improvements will be driven by such factors as the dramatic increases in the power of devices, the development of new techniques for exploiting the vast amounts of data available on the World Wide Web, and improvements in related technologies such as speech recognition. All these factors are creating a synergy that will make the next generation of natural language applications ubiquitous and indispensable parts of our lives.

References

Allen J et al (2000) An architecture for a generic dialogue shell. Nat Lang Eng 6:213–228
Alshawi H, van Eijck J (1989) Logical forms in the core language engine. In: 27th annual meeting of the Association for Computational Linguistics, Vancouver
Austin S et al (1991) Proceedings of the speech and natural language workshop, Pacific Grove
Ball CN et al (1989) Answers and questions: processing messages and queries. In: Speech and natural language: Proceedings of a workshop held at Philadelphia, Pennsylvania
Barnett JA et al (1980) The SDC speech understanding system. In: Lea WA (ed) Trends in speech recognition. Prentice-Hall, Englewood Cliffs, pp 272–293
Bohus D, Rudnicky AI (2003) RavenClaw: dialog management using hierarchical task decomposition and an expectation agenda. In: Eurospeech, Geneva
Bolt R (1980) Put-that-there: voice and gesture at the graphics interface. Comput Graph 14:262–270
Bray T et al (2004) Extensible Markup Language (XML) 1.0 (Third Edition). Retrieved November 9, 2012, from http://www.w3.org/TR/2004/REC-xml-20040204/
Brill E (1992) A simple rule-based part of speech tagger. Paper presented at the Proceedings of the third conference on Applied natural language processing (ANLC '92). Stroudsburg, PA, USA
Chomsky N (1957) Syntactic structures. Mouton, The Hague
Chu-Carroll J, Carpenter B (1998) Dialog management in vector-based call routing. In: 36th ACL/COLING, Montreal, pp 256–267
Claburn T (2009) Stephen Wolfram's Answer To Google. Information Week. Retrieved November 9, 2012, from http://www.informationweek.com/news/internet/search/215801388?pgno=1
Clark P, Harrison P (2008) Boeing's NLP system and the challenges of semantic representation. In: Semantics in text processing. STEP 2008 conference proceedings, Venice
Crouch S, Khosla R (2012) Sentiment analysis of speech prosody for dialogue adaptation in a diet suggestion program. ACM SIGHIT Rec 2:8
Dahl, Deborah A. (2006, September). Natural Language Processing: The next steps. Speech Technology Magazine, 11
Dahl DA et al (1994) Expanding the scope of the ATIS task: the ATIS-3 corpus. In: ARPA human language technology workshop, Princeton
Dahl, Deborah A., Norton, Lewis M., & Scholz, K.W. (2000, November). Commercialization of natural language processing technology. *Communications of the ACM (electronic edition)*, 43
Dahl DA et al (2011) A conversational personal assistant for senior users. In: Perez-Marin D, Pascual-Nieto I (eds) Conversational agents and natural language interaction: techniques and effective practices. IGI Global, Hershey, Pennsylvania
Erman LD et al (1980) The HEARSAY-II speech understanding system: integrating knowledge to resolve uncertainty. Comput Surv 12:213–253

Fellbaum C (ed) (1998) WordNet: an electronic lexical database. MIT Press, Cambridge, MA
Fillmore C (1985) The case for case. In E. Bach & R. T. Harms (Eds.), Universals in Linguistic Theory (pp. 1–88). New York: Holt, Reinhart and Winston
Grishman R et al (1994) Comlex Syntax: building a computational lexicon. In: COLING, Kyoto, pp 268–272
Hirschman L (1989) Overview of the DARPA speech and natural language workshop. In: Speech and natural language: Proceedings of a workshop held at Philadelphia, Pennsylvania
Hopcroft JE, Ullman JD (1987) Introduction to Automata Theory, Languages, and Computation. Reading, MA: Addison-Wesley Publishing Company
Hunt, Andrew, & McGlashan, Scott. (2004) W3C Speech Recognition Grammar Specification (SRGS). Retrieved November 9, 2012, from http://www.w3.org/TR/speech-grammar/
Jackson E et al (1991) A template matcher for robust NL interpretation. In: Speech and natural language: proceedings of a workshop held at Pacific Grove, California, 19–22 Feb 1991, Pacific Grove
Johnston M et al (2009) EMMA: Extensible MultiModal Annotation markup language. Retrieved November 9, 2012, from http://www.w3.org/TR/emma/
Jokinen K, McTear M (2010) Spoken dialog systems. Morgan & Claypool, San Rafael
Larrson S, Traum D (2000) Information state and dialog management in the TRINDI dialog move engine toolkit. Nat Lang Eng 6:323–340
Lee K-F (1989) Automatic speech recognition: the development of the SPHINX system. Kluwer, Norwell
Levin B (1993) English verb classes and alternations. The University of Chicago Press, Chicago
Liu X et al (2011) Recognizing named entities in tweets. Presented at the proceedings of the 49th annual meeting of the Association for Computational Linguistics, Portland
Marcus M (1980) A theory of syntactic recognition for natural language. MIT Press, Cambridge, MA
Marcus M et al (1993) Building a large annotated corpus of English: the Penn Treebank. Comput Linguist 19:313–330
McGlashan S et al (2004) Voice Extensible Markup Language (VoiceXML 2.0). Retrieved November 9, 2012, from http://www.w3.org/TR/voicexml20/
Minsky, M (1975) A framework for representing knowledge. In: Nash-Webber BL, Shank R (eds) *TINLAP '75 Proceedings of the 1975 workshop on Theoretical issues in natural language processing* (pp. 104–116). Stroudsburg, PA: Association for Computational Linguistics
Moschitti A et al (2011) Using syntactic and semantic structural kernels for classifying definition questions in Jeopardy! In: Proceedings of the conference on empirical methods in natural language processing, Edinburgh
Nguyen A, Wobcke W (2005) An agent-based approach to dialogue management in personal assistants. Presented at the proceedings of the 10th international conference on intelligent user interfaces, San Diego
Norton LM et al (1991) Augmented role filling capabilities for semantic interpretation of natural language. In: Proceedings of the DARPA speech and language workshop, Pacific Grove
Norton LM et al (1992) Recent improvements and benchmark results for the Paramax ATIS system. In: Proceedings of the DARPA speech and language workshop, Harriman
Norton LM et al (1996) Methodology for application development for spoken language systems. In: International conference on spoken language processing, Philadelphia, pp 662–664
Oshry M et al (2007) Voice Extensible Markup Language (VoiceXML) 2.1. Retrieved November 9, 2012, from http://www.w3.org/TR/voicexml21/
Palmer M et al (1993) The Kernel text understanding system. Artif Intell 63:17–68
Palmer M et al (2005) The proposition bank: an annotated corpus of semantic roles. Comput Linguist 31:71–105
Rabiner LR (1989) A tutorial on hidden Markov models and selective applications in speech recognition. Proc IEEE 77(2):257–286
Rudnicky AI, Hauptmann AG (1992) Chapter 10: Multimodal interaction in speech systems. In: Blattner MM, Dannenberg RB (eds) Multimedia interface design. ACM Press, New York, pp 147–171

Seneff S (1992) TINA: a natural language system for spoken language applications. Comput Linguist 18:61–86
Shannon C (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423, 623–656
Sidner CL (2004) Building spoken-language collaborative interface agents. In: Dahl DA (ed) Practical spoken dialog systems. Kluwer, Dordrecht
Standard ECMA-327 (2001) ECMAScript 3rd Edition Compact Profile. Retrieved November 9, 2012, from http://www.ecma-international.org/publications/standards/Ecma-327.htm
Taylor MM, Neel F, & Bouwhuis, DG (eds) (1989) The Structure of Multimodal Dialogue. Amsterdam: North-Holland
Telephony voice user interface conference. Available: http://www.tmaa.com/
Turing A (1950) Computing machinery and intelligence. Mind 59:433–460
Turney P (2002) Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Paper presented at the Proceedings of the Association for Computational Linguistics, Philadelphia
Van Tichelen L, Burke D (2007) Semantic Interpretation for Speech Recognition. Retrieved November 9, 2012, from http://www.w3.org/TR/semantic-interpretation/
Ward W (1989) Understanding Spontaneous Speech. Paper presented at the DARPA Speech and Language Workshop, Philadelphia
Weizenbaum J (1966) ELIZA—a computer program for the study of natural language communication between man and machine. Commun ACM 9:36–45
Wolf JJ, Woods W (1980) The HWIM speech understanding system. In: Lea WA (ed) Trends in speech recognition. Prentice-Hall, Engelwood Cliffs, pp 316–339
Woods WA (1970) Transition network grammars for natural language analysis. Commun Assoc Comput Mach 13:591–606
Woods W et al (1972) The lunar sciences natural language information system: final report. Bolt, Beranek and Newman, Cambridge, MA
Yngve VHA (1960) A model and a hypothesis for language structure. Proc Am Philos Soc 104:444–466

Chapter 5
Sequence Package Analysis: A New Natural Language Method for Mining User-Generated Content for Mobile Uses

Amy Neustein

Abstract Paradoxically, in an era when cyber-postings proliferate on the Web, much of the valuable information that can be mined from user-generated content (UGC) still eludes most mining programs. One reason this massive amount of UGC is, for all practical purposes, "lost" in cyberspace has to do with the limitations inherent in existing approaches to natural language understanding. In this chapter, I will explore how Sequence Package Analysis (SPA), a new natural-language data-mining method for text-based and spoken natural-language input, locates and extracts valuable opinion-related data buried in online postings—and makes such data available to mobile users. The SPA mining method can be used with existing SLM systems to assist in both *supervised* and *unsupervised* training. This chapter demonstrates that the advantage of SPA in such contexts is twofold: First, by breaking down unconstrained, open-ended natural-language input into relevant sequence packages, SPA can streamline the process of classifying a vast number of sentences (or spoken utterances); second, as the SPA algorithms become more robust, the process of collecting and classifying natural-language input can be automated entirely, thereby replacing human annotators with SPA-designed machine-learning. Using several examples, randomly selected from the TripAdvisor website, I illustrate how SPA can render the hidden attributes of online reviews (both positive and negative) more visible to the mobile user.

A. Neustein, Ph.D. (✉)
Linguistic Technology Systems, 800 Palisade Avenue, Suite: 1809, Fort Lee, NJ 07024, USA
e-mail: amy.neustein@verizon.net

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_5,
© Springer Science+Business Media New York 2013

Introduction

Paradoxically, in an era when cyber-postings proliferate on the Web, much of the valuable information that can be mined from user-generated content (UGC) still eludes most mining programs. In the mobile setting in particular, access to UGC may be even more critical. In Di Fabbrizio et al. (2013) the authors point out that "[c]onsumers on-the-go increasingly rely on internet search to find services and products, and on online reviews to select from among them" (p. 290).

One reason this massive amount of UGC is, for all practical purposes, "lost" in cyberspace has to do with the limitations inherent in existing approaches to natural language understanding. To wit, a semantic grammar-based system that looks to extract relevant sentences from online opinion reviews will skip over data where there is no exact match between the user's own description of a consumer product/service and the key phrases that are coded into the semantic grammar-based system. The same data may also be lost on a statistical language modeling (SLM) system, which may yield a confidence score that is too low for finding an acceptable probabilistic match between the user's description of a product/service and the corpus of annotated-training data contained in the statistical-language modeling system.

Consider just one category of UGC, for example: the ever-expanding repository of online consumer product and service reviews. This category covers a wide range of review subjects: restaurants and hotels; movies, concerts and tourist attractions; fitness spas and yoga classes; pharmaceutical products and medical devices. Proper analysis requires better natural-language methods, broad enough to recognize the diversity of expression contained in the text of such reviews. In addition, given the vicissitudes of market conditions and seasonal trends, new products and services are constantly being introduced to the marketplace, requiring the recognizer to process the new words and phrases contained in consumer reviews. In short, for a natural-language understanding program to be effective it must able to keep pace with the flexible vocabulary of user-generated content present on the web.

Dahl (2013) closely examines the various approaches to natural language understanding. She points out that speech recognizers that utilize a statistically-based approach to interpret the meaning of unconstrained natural-language input based on "the expected possibilities of words in a user's utterance, rather than grammars," are "more flexible than grammar-based recognizers for recognizing unexpected input." Nevertheless, Dahl cautions the reader about the evident challenges to building statistical systems:

> Because these are statistically-based systems, a drawback to SLM systems is that they require collection of significant numbers of the utterances that are used to train the system, up to tens of thousands in some cases. Moreover, not only must these training utterances be collected, but they must also be manually classified into their appropriate categories by human annotators. This is because the system develops the statistical preferences that it will use to categorize future utterances on the basis of human-annotated data. Training based on human annotation, or *supervised training*, is an expensive procedure. For this reason, training with little or no attention from human annotators, called *unsupervised training*, or *weakly supervised training*, is an important goal of work in this area, although the problem of effective unsupervised training is far from solved (p. 64).

In this chapter, I will explore how Sequence Package Analysis (SPA), a new natural-language data-mining method for text-based and spoken natural-language input, locates and extracts valuable opinion-related data buried in online postings—and makes such data available to mobile users. The SPA mining method can be used with existing SLM systems to assist in both *supervised* and *unsupervised* training. This chapter demonstrates that the advantage of SPA in such contexts is twofold: First, by breaking down unconstrained, open-ended natural-language input into relevant sequence packages, SPA can streamline the process of classifying a vast number of sentences (or spoken utterances); second, as the SPA algorithms become more robust, the process of collecting and classifying natural-language input can be automated entirely, thereby replacing human annotators with SPA-designed machine-learning. Using several examples, randomly selected from the TripAdvisor website, I illustrate how SPA can render the hidden attributes of online reviews (both positive and negative) more visible to the mobile user.

Background

For over a decade, my research on SPA has appeared in peer-reviewed journals and in refereed conference proceedings (Neustein 2001, 2004a, 2006a, b, 2007a, b, 2011, 2012). This work is cited by a number of AI-researchers. Those interested in data mining in call centers focus on SPA's potential to "caption the text"—that is, to find subtle features in call-center recordings such as "early warning signs of customer frustration" (Paprzycki et al. 2004). Others have noted the utility of SPA for applications other than call center operations. For example, SPA has been pointed to as possibly part of the broad spectra of "medical natural-language mining tools" that may assist in the successful classification of "affective" versus "informative" content found in health-related web postings (Denecke 2008). Finally, patent applicants have cited my publications on SPA to support their PTO (Patent and Trade Office) applications for data-mining technology (Gallino 2008; Blair and Keenan 2009).[1]

Adapting to Less-Than-Perfect Natural Speech

The basic premise of SPA is that natural language systems, instead of seeking to "train" humans to accommodate their speaking patterns to the speech interface, must be able to *adapt* to the less than perfect speech produced by humans. Speech in general is characterized by circumlocutions, ambiguities, ellipses and other vagaries that

[1] Though SPA has not yet been alpha/beta tested, that may change shortly given the emerging applications of this technology.

can render the search for keywords or key phrases by even the best robust-parsing methods now available to us an exercise in futility. SPA, however, works with the quirks and general imperfections of natural speech, unlike today's natural-language systems, which, when faced with convoluted speech, have learned the art of "sidestepping" such convolutions instead of trying to unravel their intricacies. This is not really surprising, given that natural language understanding does not mean computers truly "understand" natural-language input as humans do (Dahl 2013). That is, to make sense out of meandering, unconstrained, open-ended input, such systems normally fall back on the recognition of a key word or phrase, which can sometimes be guided by chance.[2] But what happens when a keyword or key phrase fails to show up in the convoluted speech input altogether? As one can see, such an approach to performing recognition on circuitous, winding speech input is not only far from foolproof but also fails to bring us closer to designing natural-language systems that can truly adapt to the way people speak in the real world.

SPA-Designed BNF (Backus-Naur Form) Table

In Neustein (2006b, 2007a, b, 2011) I've shown the way SPA adjusts to speech that is less than "perfect." The method is to offer a set of algorithms that can work with, rather than be hindered by, ambiguities, ellipses and other imperfections of natural language. By breaking down natural language into a series of related turns and parts of turns discretely packaged as a sequence of (conversational) interaction, I've designed a BNF (Backus-Naur Form) table consisting of 70 sequence packages. The parsing structures contained in each sequence package consist of a set of non-terminals—context-free grammatical units and their related prosodic features—for which there is a corresponding list of *interchangeable* terminals: words, phrases, or a whole utterance.

Like the BNF tables widely used to denote *syntactic* parts of natural language grammars, the SPA-designed BNF table that is used to identify conversational sequence patterns consists of parsing structures that provide for the incremental design of complex grammatical components from more elemental units. What distinguishes the SPA-designed BNF table from a conventional table, however, is that its parsing structures are not syntactic components, encompassing parts of speech and phrases, such as N, V, ADJ, NP, VP or ADJP. Instead, they are *sequentially-implicative* units, meaning that their formal grammatical representation is defined by sequence as opposed to syntax (Neustein 2001).

[2] Dahl (2013) provides an excellent example of how systems using "[t]ext classification, in combination with statistical speech recognition based on statistical language models (SLMs)," can accurately interpret what a caller is saying "even on very indirect requests":

User: "I've been on in and out of the hospital and I know I'm late on it and I'm… I'm… I'm wondering, I'm out of the hospital now and they finally took my cast off, but I still can't work and I can't walk and I'm wondering…."

Classified as "Caller would like to get an extension on paying his utility bill" (pp. 63–64).

By relying on the sequence package in its entirety as the *primary* unit of analysis, rather than on isolated syntactic parts (such as N, V, or NP), the SPA-designed BNF table is able to depict the conversational sequences actually found in natural language input. Using an SPA-designed BNF table of multi-tiered grammatical structures, many of the subtleties, convolutions and complexities of natural language can be more effectively represented. For example, a "very angry complaint" is represented on the BNF table as the normal accretion of more elemental parsing structures, such as assertions, exaggerations and declarations.

The utility of SPA is that in parsing dialog for its relevant sequence packages, the SPA-designed natural-language interface is able to extract important business-intelligence data, including some of the more subtly expressed emotional content. It can achieve this by looking at the placement, order, and arrangement of the *totality* of the context-free grammatical units and components that make up each sequence package. Furthermore, since natural speech consists more of a composite of sequences than a string of isolated keywords or phrases, it is clear that speech applications and text-analytic mining programs equipped with the kind of sequence structures illustrated in the table can better accommodate how people really talk.

SPA's Hybrid Approach to Natural Language Understanding

To identify sequence packages, SPA uses a hybrid approach. In part, SPA's method is semantic grammar-based, for those clearly defined sequence packages that contain specifically marked boundaries and specifying package properties; in part, SPA's method is statistical, using *N-grams* to depict the probabilistic occurrence of a sequence package structure when one is not so clearly defined. However, since sequence packages are both domain-independent (Neustein 2011)[3] and language-independent (Neustein 2004b),[4] the costs of using a statistical approach are not prohibitive as they are for those applications where "data changes dynamically," requiring an expanding vocabulary to accommodate the new words for each new product, as "is the case for seasonal applications" (Dahl 2013) (p. 65).

[3] Neustein (2011) states, "Sequence packages are frequently transferable from one contextual domain to another. What this means is that many of the same sequence package parsing structures (whether they are single or multi-tiered) found in call center dialog may be found, for example, in conversations between terror suspects, doctors and patients, or teachers and students" (p. 5). Similarly, many of the same sequence package parsing structures found in text-based (as opposed to spoken) natural-language input are transferable from one domain to another. Regardless of the genre of user-generated content, the same sequence package parsing structures can be found across the wide range of topics discussed in online communications, from restaurant reviews to heated political discussions.

[4] Neustein (2004b) showed that by focusing on the social organization of talk, rather than on a sentence or an isolated syntactic part, SPA may be applied to a wide range of other languages because "*all* forms of interactive dialog, regardless of their underlying grammatical discourse structures are ultimately defined by their *social* architecture" (p. 2) (emphasis in the original).

Whether a rules-based or statistical language modeling approach is used, the main focus of SPA is to accommodate to locally (contextually) produced natural-language data by mapping out the orderly sequence packages that emerge as *indigenous* to natural language (Neustein 2001), both as speech and as text . For this reason, the BNF table described above is specifically designed to capture the spoken and text-based sequence patterns which are constituted in situ; that is, within the local, situated context of the unfolding dialog or online-posting.

Methodological Origins

In constructing algorithms that portray conversational sequence patterns, SPA draws from the field of conversation analysis, a rigorous, empirically-based method of recording and transcribing verbal interaction (using highly refined transcription symbols to identify linguistic and paralinguistic features (Atkinson and Heritage 1984)) to study how speakers demonstrate, through the local design of their speaking turn, their understanding and interpretation of each other's social actions. While conversation analysis is principally directed at the study of human-human interactive dialog in both formal settings—such as courtrooms, classrooms and hospitals—and informal everyday conversations, more recently some conversation analysts have applied particular aspects of this important body of research to the study of human-computer interaction.

For example, in Moore et al. (2011) and Moore (2013) Moore and his colleagues have examined online query searches by relating some of the basic principles of ethnomethodology and conversation analysis to this area of study. Moore (2013), in studying how referential practice is organized in the context of search-engine interactions, showed how certain interactions with a GUI uncannily resemble human-to-human conversation. Pointing to the conversation analytic finding of Sacks and Schegloff (1979) that speakers display two structural preferences when making reference to persons in telephone calls, one for "minimization" (the use of a *single* term, such as a first name) and the other for "recipient design" (that the term is recognized by the other speaker), Moore showed that web searchers, likewise, show a preference for formulating their queries by using short, simple terms (such as names) for the entity that constitutes their online search.

Building on this argument, Moore revealed how even the nature of the repair work that occurs in conversations when reference terms are not recognized by the other speaker (such as the name of the third person mentioned in the conversation), closely resembles the repair work performed by web users when an online query search fails to bring up the desired information. In a conversation, as Moore points out, "sometimes the recipient cannot be expected to recognize the name of the third person (e.g., Daniel). In such cases, the preference for minimization is relaxed just enough to enable the recipient to achieve recognition through combined references forms or descriptions (e.g., Daniel, the guy who cuts my grass)" (p. 262).

However, as soon as recognition is achieved in conversation, those lengthier, more roundabout descriptions are immediately abandoned for the short, single reference

terms because "speakers seek mutual recognition with the fewest words or least amount of interactional work possible" (p. 262). Moore showed that the same holds true for search engine interactions. That is, after names fail to bring up the desired search results and users must, instead, resort to generic descriptions, users immediately abandon those lengthy generic descriptions, once the correct name for the search item is (apparently) learned, in favor of using the correct name in all of their subsequent online searches.

In fact, since conversation analysis is informed by ethnomethodology—the study of how social interactants accomplish the situated production of social order in their day-to-day activities—I suspect that we will eventually realize that many of the research findings of conversation analysts detailing "how speakers locally organize *talk-in-interaction* through generic, but situated sequential practices" may be applied to the study of some of the nonverbal ways that social interactants "locally achieve order in concrete social settings" (Moore 2013) (p. 263). Users' in situ interactions with search engines, as discussed above, serve as a good example of the application of conversation analysis to text-based interactions.

Certainly, the application of ethnomethodology to better understand, in more general terms, human interactions with GUIs does not present a novel concept. Lucy Suchman (1987) argued nearly three decades ago in *Plans and Situated Order* that system designers must be cognizant of the fact that user interaction with machines, as with humans, is a characteristically ad hoc, situated achievement that does not lend itself to an a priori designation of plans and goals. Hutchby and Wooffitt (1998) point out that "Suchman's work has had an important impact on the field of system design. Not only did it propose a strong critique of the user as plan-following and goal-seeking, but it introduced the significance of *conversation analysis* … to a community of system developers" (p. 243) (emphasis supplied).

As we have seen, the methodological groundings of SPA provide a rich, substantive basis for formulating a new natural language method that is in synchrony with the conversational sequence patterns of both spoken and text-based natural language input. By studying natural language input as it is produced in situ by tweeters, bloggers, and social networkers (and anyone else who fits into the more general category of online reviewers or posters), SPA equips natural speech systems with a keener understanding of the messages conveyed in user-generated content posted on the Web. In practical terms, what this means is that an SPA-driven natural-language system could mine the web for valuable feedback on consumer products and services that would have otherwise remained hidden, as well as provide critical systems with homeland-security intelligence data that could have all too easily been overlooked by conventional mining programs (Neustein 2006b).

Methodological Caveats

As we have seen in the prior section, there are benefits to drawing from the conversation analytic literature for the design of natural-language systems that can accurately represent the dynamic, in situ organization of human communication, whether it takes the form of spoken language input or online-community postings. Nonetheless,

in the interest of fairness, I will take a moment here to present the views of those within the conversation analytic field who have objected to the derivation of programming rules from what has been learned about the systematic and orderly features of human communication. After all, the caveats they pose can only serve as helpful reminders of the obstacles that must be rigorously overcome:

1. In Button et al. (1995) the authors assert that "inferential possibilities of a sentence" are refractory to programming rules (p. 176). They use this argument to support their objection to the use of conversation analysis as the source of programming rules.
2. In Button (1990) the author asserts that the rules operating in conversation are not "codifiable" or "reducible to an algorithm" (p. 84).
3. In Schegloff (1992) the author points out that "possible [turn] completion is something projected continuously (and potentially shifting) by the developing course and structure of the talk," (p. 118) rendering human dialog too unpredictable and changeable, moment to moment, to be reduced to a set of programming rules (Button and Sharrock 1995).

Here are the principal counter-arguments posed to such caveats:

1. Gilbert et al. (1990) contradict those who assert that human dialog is resistant to programming rules simply because the meaning of utterances present limitless possibilities for interpretation depending on context. They start by pointing out that speakers, in their day-to-day interactions with other speakers, routinely work in situ to achieve order by redressing the contextually-dependent indigenous meaning of utterances so that meaning is not left entirely open-ended and subject to manifold interpretations: "… [because] the meaning of specific terms or expressions is not fixed, as in a dictionary definition, nor computable using simple rules of deduction, but dependent on the context in which the item is embedded [t]he hearer has to *work actively to find a meaning for the term which makes sense within that context*" (p. 254) (emphasis supplied).
2. In Gilbert et al. (1990) the same authors, describing this orderly way in which interlocutors redress the open-ended possibilities for interpretation caused by contextually-dependent meaning, draw an analogy to computing. They show that just as in human-to-human interactions, speakers overcome the problem of context-dependent meanings by treating new material as an instance of a presupposed underlying pattern against which new material can be interpreted, in computational modeling "the grammar a chart parser operates on will have alternative 'patterns' against which the input can be matched" (pp.255–256).
3. Hirst (1991) who, more than two decades ago, espoused the use of conversation analysis in natural speech systems, has stated: "it is clear conversation analysis must have a role in Natural Language Understanding because there is a sense in which [conversation analysis] is just a small sub field of artificial intelligence" (p. 225).
4. Hutchby and Wooffitt (1998) point to the impoverished methods of those who design interactive systems without a full appreciation of conversational analytic findings: "there has been an unfortunate tendency to discuss aspects of conversational organization … in the abstract, removed from empirical materials"

(pp. 244–245). It is further believed "that in order to design computer systems which either simulate, or more ambitiously reproduce the nature of human communication, it is necessary to know about the ways in which everyday (conversational) interaction is organized" (p. 241).

Yet so far, with all the pronouncements about the benefits of using conversation analysis for computer modeling of natural speech, no one has introduced a detailed approach that applies conversation analysts' empirical findings on the generic orderly sequences that emerge in situ in *talk-in-interaction* to successfully build simulacra for human dialog. This is where SPA finds its purpose: to provide an algorithmic framework, bridging the empirical research findings of conversation analysts with the design constraints of natural language modeling. The next section provides illustrations of how SPA extracts useful data often obscured in user-generated content.

Illustrations of Indigenous Sequence Packages

Finding the Hidden Negative Attributes in Online Consumer Reviews

In this section, I show how an SPA mining-program can be applied to consumer reviews of a fast-food restaurant, which may prove critical in a mobile setting given that users "on-the-go" may be more likely to stop at a fast-food place than to eat at a restaurant that would require a reservation. I randomly chose to examine two reviews posted in the past 4 months for the "Falafel Drive-In" in San Jose, California. These reviewers were found on TripAdvisor, a popular web site for consumer reviews of restaurant, hotel and travel services.

Example One

Below is the unedited text of a consumer review posted to TripAdivsor. The reviewer's punctuation, including use of n-dashes, is reproduced below just as it appears in the online posting.

TripAdvisor

"Falafel Drive-In" in San Jose

"Excellent Falafels and Shakes!"
Reviewed August 4, 2012 (5-star rating)
(Value, service, atmosphere and food: not separately rated)

I've been here 4–5 times at least and I never leave disappointed. Parking can be tough during the lunch crowd but it is totally worth it. There is typically a line—a good sign in my opinion! They have a small indoor seating area but tons of outdoor seating. The falafel is excellent. I always ask for a side of their hot sauce because it's

that good! The falafel combo deal is great because it is cheap and it comes with their fantastic banana shake! The banana shake is the best I've ever had! They do not accept credit cards, only debit and cash so come prepared. This place is a must if you leave [sic] in San Jose! Excellent, just excellent!

Sequence Package Parsing Structures
<Opening Endorsement> *"I've been here 4–5 times at least and I never leave disappointed"*
<Complaint/Disclaimer (Parking)> "Parking can be tough during the lunch crowd but it is totally worth it"
<Complaint/Disclaimer (Waiting)> "There is typically a line – a good sign in my opinion!
<Complaint/Disclaimer (Seating)> "They have small indoor seating but tons of outdoor seating"
<Opinion Review> *"The falafel is excellent. I always ask for a side of their hot sauce because it's that good! The falafel combo deal is great because it is cheap and it comes with their fantastic banana shake! The banana shake is the best I've ever had!"*
<Complaint/Disclaimer (Payment)> "They do not accept credit cards, only debit and cash so come prepared"
<Closing Endorsement> *"This place is a must if you leave* [sic] *in San Jose! Excellent, just excellent!"*

Analysis

In this restaurant review, cited above, though the consumer gave the restaurant a 5-star overall rating, she subtly pointed out a number of problems (difficulty parking, waiting in line, limited indoor seating and payment restrictions) which may be of importance to other consumers in deciding whether to patronize this drive-in eatery. Conventional mining program that extract relevant sentences and collocated words and phrases would not be readily able to detect opinion data when it is cloaked in this way. In contrast, as will be shown below, the sequence package structures contained in this online review are of such a generic kind that the opinion data, no matter how subtle or indirect, would not escape an SPA-designed mining program. For the purpose of this illustration, and the subsequent illustrations presented below, I will concentrate on SPA components, which are the larger parsing structures, rather than their smaller units. Since all components are derived from their smaller parts, an SPA mining program would naturally have both the smaller structures from which the larger ones are built.

What emerges indigenously in this online review is a sequence package known as a *contrastive pair* (Neustein 2001). The type of contrastive pair that is found here is the "complaint/disclaimer" pair. That is, each time a complaint is made it is immediately followed by some sort of a "disclaimer." The disclaimer may take the form of a justification, rationalization or solution. Whatever form it takes, its effect is the same, as it serves to "downgrade" or nullify the complaint.

Figure 5.1 below shows the series of four complaint/disclaimer contrastive pairs found in this online review. Three of these contrastive pairs are consecutive, sandwiched between the opening endorsement and the opinion review, while the fourth one appears immediately after the opinion review. The reason for the appearance of the last complaint/disclaimer pair after the opinion review and not before it (as was the case with the prior three complaint/disclaimer pairs) is mainly topical. That is, the last complaint/disclaimer pair refers to post-eating conditions in the restaurant (i.e., payment), whereas the first three pairs are relevant to conditions prior to eating (parking, waiting in line, and seating).

Figure 5.2 below shows the grammatical structure of the second part of the contrastive pair, which begins with a concessive connector ("but", "so," "n-dash"), otherwise referred to by conversation analysts as a "contrast marker," followed by an idiomatic expression or metaphor. (Here, idioms are defined rather broadly to include banalities, platitudes and clichés that serve as a "shorthand" way of getting the message across—in which their connotative meaning is not necessarily deducible from the individual words that make up the idiom.) Since conversation analytic studies have shown that idioms and metaphors serve special purposes, we can see from this posting that the online reviewer's use of these expressions has not occurred arbitrarily. Metaphors have been found to be helpful in achieving topic transition in conversation (Drew and Holt 1998). In this example, "tons of seating" is followed by a topic transition away from the series of complaints to the rendering of an opinion review of the drive-in ("The falafel is excellent …).

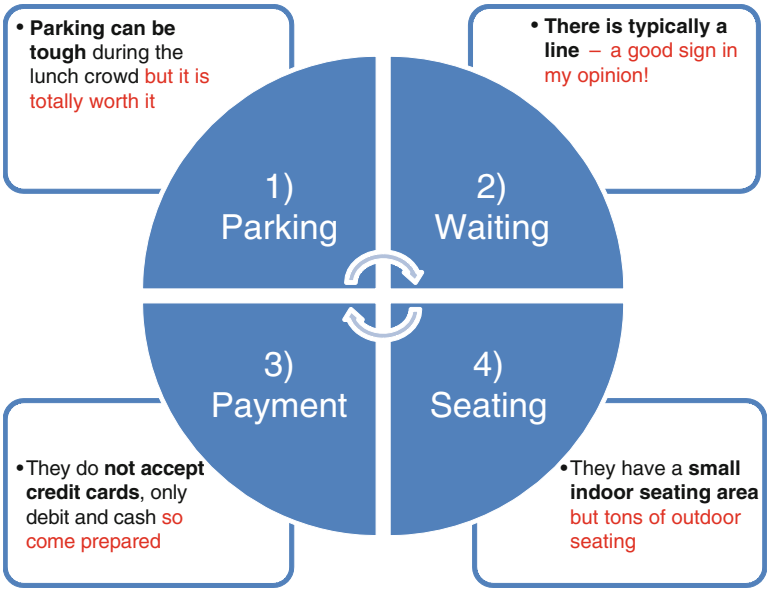

Fig. 5.1 *Complaint/Disclaimer Contrastive Pair* (each contrast utterance begins with a concessive connector, referred to here as contrast marker: "but," "so" or an "n-dash")

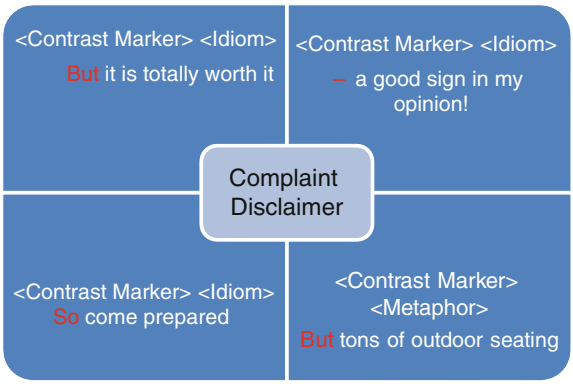

Fig. 5.2 Parsing structures of second pair part of complaint/disclaimer contrastive pair: contrast marker followed by idiom or metaphor

Idioms fulfill another function as well. They achieve indefeasible arguments because the idiom, itself a product of a culturally established "stable body of knowledge," is not subject to challenge (Drew and Holt 1988; Pomerantz 1986; Torode 1995). Consequently, when there is a dispute, idioms can be used rather skillfully to forge consensus. In Pomerantz (1984) the author examined dialog in which one of the speakers reverses her position on a sensitive matter by supporting "the newly affirmed position with … (an) aphorism" (p. 161). Similarly, in this TripAdvisor posting, the reviewer uses an idiomatic expression each time she seeks to nullify or disclaim her criticism of the falafel joint so as not to appear to be in a dispute over the services and conditions of this place.

N-grams for Sequence Packages

Contrastive pairs, such as a complaint/disclaimer, would be spotted by an SPA recognizer, using *N-grams* to spot the collocation of the first pair part vis-à-vis its attendant contrastive-pair second pair part in much the same way that statistical models have been trained to look for a contiguous sequence in the form of *bigrams*, *trigrams*, or *N-grams*, "techniques that automatically produce large numbers of collocations along with statistical figures intending to reflect their relevance" (Smadja 1991) (p. 279). Thus, following the approach of statistically-based language-modeling systems, SPA relies on *N-grams* to produce the statistical probability for the occurrence of collocated sequence package structures, such as complaints and their disclaimers, from which the system can then extract the more subtle aspects of consumer reviews that may be hidden from conventional recognizers.

In essence, SPA systems perform a type of "robust parsing," but rather than parse spontaneous speech for its "individual [semantic] segments that make sense within the defined task" (Pieraccini 2012) (p. 163), SPA parses the natural-language input for sequence package structures that are relevant to the defined task, such as the posting of online reviews of products and services. In such postings, it is fairly common to find sequence packages of complaint/disclaimer contrastive-pairs interspersed among the

Fig. 5.3 *Opinion sequences*

more rudimentary *opinion sequences*. In Fig. 5.3 above the three variants of opinion sequences: opening endorsement, opinion review and closing endorsement are shown.

Complaint Sequences Versus Opinion Sequences

It is easy to see why this series of complaint sequences is buried in this online posting. First of all, given the fact that many reviewers are reluctant to criticize a recognized establishment, negative feedback often takes the form of an indirect statement. Producing a disclaimer—joined to the complaint by a concessive connector or contrast marker—immediately after the complaint provides a diplomatic way to retreat or withdraw from one's position. Though a mining program can't read the reviewer's mind, her production of a series of complaints, each of which is disclaimed immediately afterwards, demonstrates the reviewer's predisposition to minimize anything problematic about the "Falafel Drive-In." If the reviewer retreats, then how can a mining program inform other consumers about the restaurant's downside? That is, when such complaints are routinely being minimized by online reviewers seeking to diminish the importance of their own uncomplimentary feedback, how can this information become available to other customers who may benefit from knowing ahead of time the drawbacks of the enterprise?

Though this presents something of a conundrum, an SPA-designed mining-program would try to solve the problem by first taking into consideration that opinion sequences (opening endorsement, opinion review and closing endorsement) are themselves indigenous features of the online posting. That is, the opinion sequences used to appraise this restaurant are produced in situ—so much so that the reviewer's consistent retreat

from each complaint she raised contributed to her 5-star overall rating for the eatery (though she failed to individually rate the four specific categories on *TripAdvisor*: service, food, value, atmosphere). Her rating was matched by her highly favorable review headline (appearing in TripAdvisor's "title of your review" box), which read "*Excellent Falafels and Shakes*!" A *cycle* matrix diagram has been used in Fig. 5.1 to demonstrate that each complaint/disclaimer contrastive pair is no more or less than part of a cycle that informs the overall restaurant rating as an in situ achievement.

Thus, by examining the indigenous arrangement of sequence packages in natural-language communications, the mining program would be able to detect how the *superlative* rating was arrived at in the first place. In this case, it was the result of the reviewer's disregard for the concerns that she herself raised about parking, seating capacity, waiting in line and restricted payment methods. True, one might alternatively argue that such concerns did not trouble this reviewer in the first place, especially given her 5-star rating followed by her superlative assessment. We may never know exactly what was in this reviewer's mind. But that is not to say that the issues she raised would not have been important to another consumer who may have shied away from a restaurant with parking problems, long lines, limited indoor seating and no credit card payments accepted.

All in all, the reviewer's backpedaling from her complaints and her resultant provision of an outstanding overall rating should not preclude the mining program's ability to extract what might be, to mobile users, invaluable information in their search for a fast-food restaurant to have a quick meal. At the very minimum, what this example shows is that if we unravel these indigenous sequence packages, the kernels of data that have become submerged in the convolutions of natural-language postings can be brought to the surface and made available to mobile users.

Example Two

Below is the unedited text of a consumer review posted to TripAdivsor. The reviewer's punctuation, including the use of elliptical dots, is represented below just as it appears in the online posting.

TripAdvisor

"Falafel Drive-In" in San Jose

"Great food!"
Reviewed April 22, 2012 (5-star rating)
Value, service, atmosphere, and food (5-star rating on all features, except atmosphere which was given four stars).

We read all the great reviews and decided to give it a try. We only had a short time for lunch and this was perfect. Both of us had the falafel sandwich; I had a banana shake, my husband a vanilla shake. Well, everything was great. The ingredients were fresh. The sauce was yummy. The sandwich fell apart after a while but we just continued eating it with a fork.... No problem. The service was very fast.
Great place. We recommend it!!

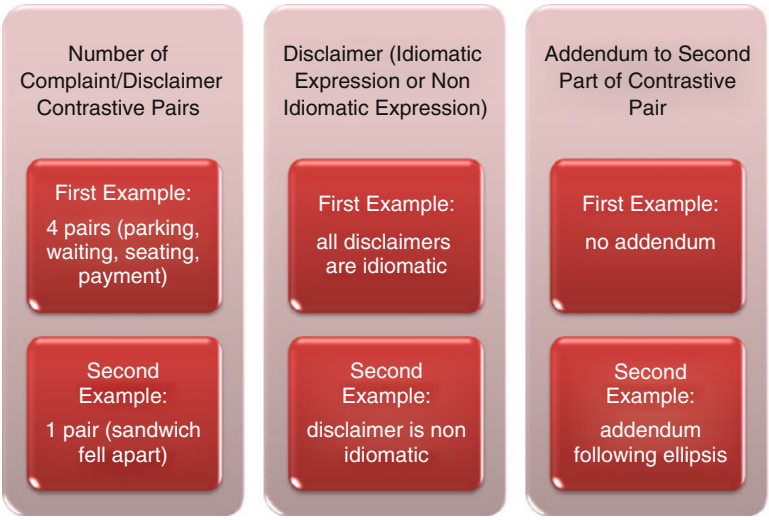

Fig. 5.4 Comparing the differences in sequence package structures between the first example and the second example

Sequence Package Parsing Structures
<Opening Third Party Assessment> <Personal Narrative> <Endorsement>
"We read all the great reviews and decided to give it a try. We only had a short time for lunch and this was perfect."
<Personal Narrative>
"Both of us had the falafel sandwich; I had a banana shake, my husband a vanilla shake."
<New Topic Interjection Marker> <Opinion Review>
"Well, everything was great. The ingredients were fresh. The sauce was yummy."
<Complaint/Disclaimer> <Ellipsis> <Formulation>

"The sandwich fell apart after a while but we just continued eating it with a fork.... No problem"
<Opinion Review>
"The service was very fast."
<Closing Endorsement>
"Great place. We recommend it!!"

Analysis

This online review, while providing a 5-star rating (just as in the first online review), nonetheless demonstrates a variation on the sequence package arrangement found in the prior example. Below are the differences, which have been outlined in Fig. 5.4.

1. There is only one complaint/disclaimer contrastive pair in the second example (*"The sandwich fell apart after a while but we just continued eating it with a fork.... No problem"*) as opposed to a series of four complaint/disclaimer pairs found in the first example;
2. The disclaimer in the second example, unlike in the prior review, does not take the form of an idiom or metaphor; instead, following the contrast marker, a straightforward complaint resolution statement is provided ("but we just continued eating it with a fork");
3. A special addendum to the complaint/disclaimer contrastive pair is found in the second example, but not in the first. The addendum consists of a "formulation"—a grammatical device, closely studied by conversation analysts, that allows a speaker (or an online reviewer in this instance) to use some part of the dialog (or posting) to "formulate" or "sum up" the activity he/she is presently engaged in (Heritage and Watson 1979). The formulation, which takes the form of an idiomatic expression ("No problem"), permits the reviewer to "sum up" her complaint as something that is *not* important in the least.

It is interesting to note that the reviewer placed an ellipsis, a series of three dots (…) right before she "summed up" her disclaimer with the use of an idiomatic expression: "but we just continued eating it with a fork.... No problem." In fact, given that in all natural-language communications order is achieved in situ, that is, in the local, concrete setting where the communication takes place, it was neither arbitrary nor accidental that a formulation was produced immediately following the ellipsis.

The warrant for this is as follows: because an ellipsis conveys an *unfinished thought*, one which allows readers to project their own thoughts into the omission represented by the ellipsis, it would have been risky to leave it to the reader to determine whether using a fork to eat a sandwich that has already disintegrated represents a viable solution to the problem of the sandwich having fallen apart. By employing the grammatical device of "formulation," and in particular an idiomatic expression which works to ensure agreement to something that may be open to dispute, the online reviewer was able to effectively seal up the open-endedness of her complaint-disclaimer ("but we just continued eating it with a fork … No problem").

There is yet another reason for the appearance of this particular sequence package design, consisting of a formulation appended to the disclaimer (creating a stronger and more definite retraction than disclaimers that are not followed by such formulations). The reviewer's use of this particular sequence-package design shows that she may be exercising caution when providing any sort of negative feedback about the eatery, most likely because from the very beginning this reviewer knew she was assessing a well-known San Jose restaurant that had already received so many laudatory reviews on TripAdvisor.

Unlike the prior reviewer, who announced that she had been to this restaurant a number of times before and was always pleased (*"I've been here 4–5 times at least and I never leave disappointed"*), the second reviewer acknowledged the stream of laudatory reviews and that her position was that of a novice: *"We read all the great reviews and decided to give it a try.*" By placing herself as a "newcomer" to this

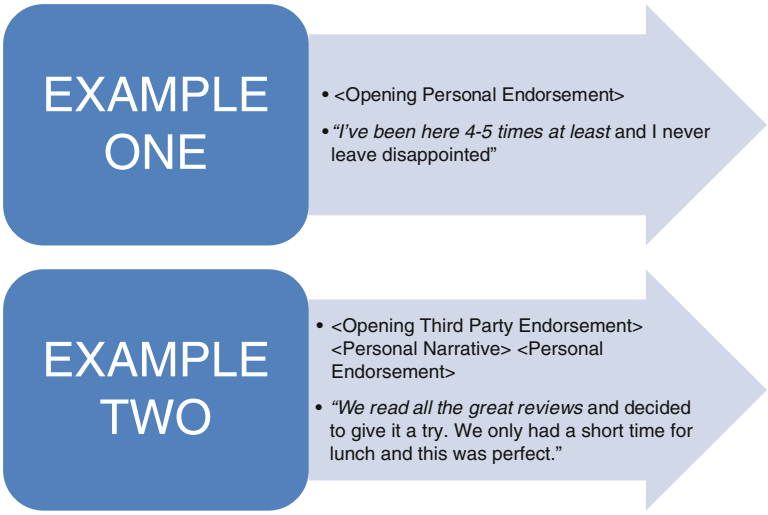

Fig. 5.5 Comparing the sequence package parsing structures used by the novice versus the seasoned visitor to this restaurant

well-known establishment—and readily acknowledging the praiseworthy reviews already posted by the other patrons—she implicitly set up her review to be measured or weighed against her (virtual) community of peers who had already supplied theirs. Figure 5.5 above shows the stark contrast between these two reviewers—one a novice, the other a regular—mapped out in the sequence package parsing structures that make up the opening statement of each review. To wit, the first example begins with a "personal endorsement" whereas the second begins with a "third party endorsement."

In actuality, even though online communities are virtual (and mostly anonymous: reviewers rarely use their real names, or if they do they do, usually their first names only), we cannot presume that the same sort of peer pressure found in real (non virtual) communities doesn't exist in virtual ones. In fact, many of the same social constraints found in real communities may be found in virtual ones as well. For example, sometimes peer pressure is not intended to forge consensus, expressed in the sharing of the opinions of others, but the exact opposite. We see this in those online postings from reviewers who, as *non locals*, seek to reinforce their position as "outside" the virtual community of local reviewers. In such cases, the leitmotif of their review postings can be that of noticeable "disagreement," as to opposed "consensus," with the prior online reviewers. Here is a brief example of such a review posting from a Houston couple of the San Jose-based Falafel Drive-In.

"I don't get it …"
Reviewed June 28, 2012

We drove directly here from the San Jose airport because of all the raving [sic] reviews. But to us it was just average, if this is the best falafels in town, then y'all

need to visit Houston, Texas. This food would not make the top. The falafels were burned on the outside and dry, and the banana shakes were small for the price. The food was just OK but really not worth raving about, for San Jose it is fantastic, for the rest of us … average.

The post of this Texas couple, unlike that of the San Jose local who displayed a preference for camaraderie with her local online community, the Houstonians have openly challenged the complimentary consumer reviews that had appeared on TripAdvisor: "We drove directly here from the San Jose airport because of all the raving [sic] reviews. But to us it was just average." Their opening statement about their discordant review is immediately followed by a litany of graphic complaints: "the falafels were burned on the outside and dry, and the banana shakes were small for the price." (Note that there are no complaint/disclaimer contrastive pairs here, which, as we've seen in the prior examples, serve to minimize or nullify the complaint.) The Houstonians conclude their review with a reconfirmation of their opinion of this eatery, one antithetical to the opinions held by the San Jose locals: "The food was just OK but really not worth raving about, for San Jose it is fantastic, for the rest of us … average."

Returning to the review of the San Jose couple discussed above, one can see that by virtue of their mention of the other "great reviews" of the Falafel Drive-In, the online reviewer immediately set herself up for a *test* as to whether she would be able to follow in the footsteps of her virtual community members, who had already supplied a number of online reviews praising this enterprise. For this reason, we are able to understand why it is that when she reported that her sandwich "fell apart"—a complaint that may be of particular interest to someone on-the-go who stops by a drive-in to grab a sandwich for eating either in the car or while hurriedly walking back to the office—she had to do serious repair work to back away from a complaint that would have put her at odds with the "great reviews" already posted by her virtual peers. The grammatical device of "formulation" produced as an addendum to the complaint/disclaimer contrastive-pair served as an effective way of backpedaling from her complaint.

In short, when comparing the reviews of the San Jose online reviewers—one a newcomer to the restaurant, the other a seasoned patron—we see some marked variations in sequence package design. However, regardless of which parsing structures appear in the online posting, one should note that sequence packages always emerge indigenously as features of the *locally* achieved order of natural-language website postings, whatever the variations in their design. By paying attention to these fine points, such as the posture of the "opening endorsement" and what it clearly conveys about the reviewer's status in the virtual community of online-opinion makers, a program would be better able to interpret/process the ensuing review.

All in all, mining programs that are directed to look at sequence package data for extracting some of the more subtly reported opinion-related information may prove quite useful to the mobile user. After all, why shouldn't a mobile user who doesn't have the luxury of reading through all the postings be forewarned about the downside to this major restaurant fixture in San Jose? Sandwiches falling apart, the long lines to get into the restaurant, a paucity of indoor seating, parking difficulties, and

their failure to take credit cards, are just a few of the negative features that may be extracted from these postings. This is not to say that mining programs should slant their findings toward negativity. According to the reviews, the Falafel Drive-In's food is by and large exceptionally good, which is certainly important for the mobile user to know. But the restaurant's less attractive features are also important in helping the mobile user to make an informed decision about stopping in for a falafel and shake while on-the-go.

While the two examples above explored how *negative* attributes may be hidden in online postings, the next section will show how *positive* attributes of an enterprise may be similarly hidden in user-generated content.

Finding the Hidden Positive Attributes in Online Consumer Reviews

Using as a data sample an online review of a New York City hotel, I show how hidden positive attributes may be extracted from such a review. The review had: (1) a critical review headline; (2) a weak rating score; and (3) some strongly pejorative descriptions of the consumer's experiences at the hotel. Nevertheless, the review also suggested some of the more desirable features of this hotel—desirable location, good air conditioning in the room, a spacious room with a very comfortable bed, a good discount on room rates, and very quick access to elevators. Those features were buried in this ostensibly negative review. For mobile users, in a rush to find a decent hotel in New York City, a program that could extract the *positive* attributes from this online posting (such as a good room size, comfortable bed, etc.), despite its appearance as a *negative* review, would be very helpful to a user in making the right decision.

Applying the same approach as I did with the analysis of the San Jose restaurant reviews, this section examines the sequence package design-features that emerge in situ in the online posting about the New York City hotel.

For the purpose of exploring sequence package arrangements that show the hidden positive attributes in online reviews, it really doesn't matter whether we draw on consumer reviews of restaurants, hotels, vacation resorts, car rental companies, cell phone services, or any other kind of consumer product or service; the sequence-package parsing structures are generic features of natural-language communications. They can be found across most, if not all, subject domains. It is their domain-independence, as pointed out earlier, that allows their transferability from one contextual domain to another. However, in contrast to restaurant reviews, hotel reviews may entail a lower occurrence of positive-endorsement parsing structures that show agreement with the favorable reviews of prior reviewers. The reason for this is that most hotel reviewers, unlike restaurant reviewers, are simply "passing through" an area on business or vacation, which means they are not as likely to feel peer pressure to concur in the opinion of the other online reviewers—as we've seen, from the examples above, where a reviewer is a permanent member of a (virtual) community (such as a resident of San Jose). The only cases in which this distinction

may *not* apply occur when hotel guests make a certain geographic spot their regular vacation destination. In such cases, the reviewers may tend to behave as permanent members of their (virtual) community of peers, rather than as onlookers. This would produce a higher rate of positive endorsements in their reviews, because as "permanent" community members they are understandably less eager to disagree with their fellow community members' previously published online reviews.

Example

Below is the unedited text of a consumer review posted to TripAdvisor. The reviewer's punctuation, including the use of elliptical dots, is represented below just as it appears in the online posting.

TripAdvisor

The New York Helmsley
(Manhattan)

"Poor customer service"
Reviewed 21 July 2012 (3-star rating)

Good location, but currently being renovated so you will have to excuse the untidy appearance of the hotel and the downstairs/reception area. I think this is probably reflected in the current price. Rooms are spacious, air conditioning effective and the beds very comfortable. However, I must admit to being a little disappointed by the attitude of the staff, which was churlish at best. When being dropped off outside the hotel. I found myself subjected to a fairly aggressive verbal assault from the taxi driver who seemed to think the tip I had offered was insufficient. At this point the bell boys (engaged in conversation with other cab drivers and passers by) made no attempt to intervene or help us with our bags. Would this have happened at other NYC hotels I have stayed at - I think not. Didn't get much better at check in. So all in all, not great first impressions. Elevators are very quick though....

Sequence Package Parsing Structures
<Compliment/Attenuation>
Good location, but currently being renovated so you will have to excuse the untidy appearance of the hotel and the downstairs/reception area.
<Post-Attenuation Analysis>
I think this is probably reflected in the current price
<Compliment/Attenuation>
Rooms are spacious, air conditioning effective and the beds very comfortable. However, I must admit to being a little disappointed by the attitude of the staff, which was churlish at best.
<Expansive Narrative Complaint>
When being dropped off outside the hotel I found myself subjected to a fairly aggressive verbal assault from the taxi driver who seemed to think the tip I had offered was insufficient. At this point the bell boys (engaged in conversation with other cab drivers

Fig. 5.6 Contrastive pairs found in New York Helmsley hotel review, consisting of two compliment/attenuation pairs and one negative endorsement/concession pair

and passersby) made no attempt to intervene or help us with our bags. Would this have happened at other NYC hotels I have stayed at - I think not. Didn't get much better at check in.
<Closing Negative Endorsement/Concession>
So all in all, not great first impressions. Elevators are very quick though....

Analysis

As shown in Fig. 5.6 above, this review contains three contrastive pairs. The first two consist of compliment/attenuation pair types, and the third consists of a negative endorsement/concession pair.

However, in contrast to the examples of the restaurant reviews presented above, the second pair parts, as shown in this current example, do not consist of disclaimers—which serve to withdraw, cancel or nullify the first part of the pair—but, instead, of both attenuations and concessions that do not entirely negate the import of the first pair part, serving rather to alloy or lessen its potency. As such, the reviewer's act of following his positive assessment of the hotel ("good location") with an attenuation ("but currently being renovated so you will have to excuse the untidy appearance of the hotel and the downstairs reception area") didn't invalidate the reviewer's positive assessment but rather weakened it instead. Had the reviewer said something like, "good location at the center of town but the noise is bothersome, making it really difficult to sleep at night," that would have consisted of a disclaimer rather than an attenuation because it would have struck right at the basis of the compliment.

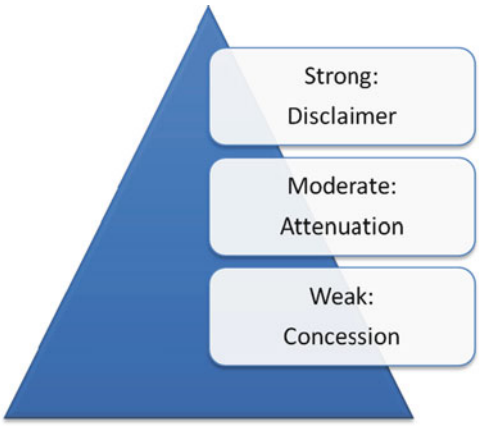

Fig. 5.7 Hierarchical arrangement of the interactive import of the second pair part in negating or diminishing what has been produced by the first pair part

In short, the main difference between disclaimers and attenuations or concessions is that while the disclaimer strikes *topically* at the first pair part head-on, in an attempt to nullify it, the attenuation or concession strikes *more generally than topically* at the source of the first part of the contrastive pair. Moreover, it doesn't much matter whether the first pair part is a complaint, as we saw in the reviews of the San Jose restaurant above, or a compliment, as we see in the present example of a New York City hotel review. For that matter, the first pair part could be any speech act, and the presence of an attenuation as the second pair part will primarily serve to weaken or lessen, as opposed to canceling or nullifying, what appears in the first pair part.

Such contrastive pairs may be arranged in a hierarchical structure based on the relative strength of the second pair part, as shown in Fig. 5.7 above. A disclaimer, which serves to directly challenge what has been produced in the first pair part (that is, to pose a challenge on the same topic that was the subject of the first part), would be rated as stronger than an attenuation or concession.[5] Figure 5.7 also shows that concessions appear even lower in the hierarchy than attenuations because they are weaker. As explained below, a comparative analysis of the sequence package arrangement of the parsing structures found in both attenuations and concessions

[5] There are occasions when the second pair part of the compliment/attenuation contrastive pair, although not a direct topical challenge to the first part of the pair, might appear strong in content even though the interactive import is still weak. For example, a person might say, "The hotel room service was exquisite though the air quality was so poor I had an asthmatic attack and had to be rushed to the hospital." In such an extreme case the second pair part is so off- topic that what would usually follow is an addendum to the contrastive pair which would try to resolve the incongruity between the parts. Sometimes humor is invoked, as in "Go figure, you get this great room service but you end up in the hospital from pollution!" Using the SPA approach to analyze natural-language communications, one would look for the sequentially-implicative units in this example. The criticism – although it entailed alarming content (landing in the hospital) – would lack potency to "disclaim" or negate the source of the compliment (which is about good room service), since its incongruence with the first pair part means it doesn't strike topically at the source of the compliment. It is therefore considered interactionally "weak" even though the content, taken outside of the sequential arrangement, might make it appear strong.

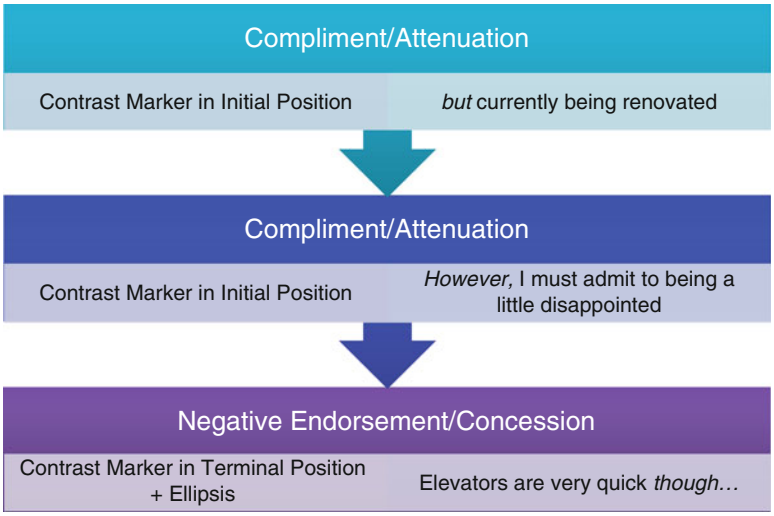

Fig. 5.8 Sequential placement of parsing structures of contrastive pairs

helps to explain the difference in the effectiveness of the second pair part in lessening what has been produced by the first pair part.

Figure 5.8 above shows the sequence-package parsing-structures appearing in the TripAdvisor review of the *New York Helmsley* that help to distinguish between an attenuation and a concession. For the purpose of this analysis, I am concentrating on the more elemental units of these sequence packages—namely, the *contrast marker* ("but", "however," and "though") and the *ellipsis* in the form of three dots (…)—upon which larger sequence package parsing structures are built.

When examining the contrast marker, what becomes important to note is the *sequential placement* of the contrast marker and not its syntactic form. In both the compliment/attenuation pairs, as seen in Fig. 5.8, the contrast marker can be found in *initial* position, which serves to introduce the attenuation which constitutes the second part of the compliment/attenuation contrastive pair. In contrast, in the (closing) negative endorsement/concession pair, the contrast marker is found in *terminal* position, following the concession, which serves to *retrospectively* create a downgrading or diminution of the negative endorsement.

In general, in gauging how strongly a speaker or writer feels about a particular topic (itself a specialized area known as "sentiment analysis"[6]), SPA pays particularly

[6] The body of research that explores reviewer attitudes in great detail is known as "sentiment analysis." Dahl (2013) explains that "[t]he goal of sentiment analysis is to characterize the speaker or writer's attitude toward the topic of the text. As reviews of products and businesses proliferate on the Web, companies that are interested in monitoring public attitudes about their products and services are increasingly turning to automated techniques such as sentiment analysis to help them identify potential problems. Sentiment analysis tries to classify texts as expressing positive, negative, or neutral sentiments, and can also look at the strength of the expressed sentiment" (p. 63).

close attention to whether a contrast marker appears in initial position, introducing the second part of the contrastive pair, or in terminal position, immediately following the second part of the contrastive pair. When contrast markers appear in initial position, as opposed to terminal position, they have a much stronger effect in negating what has occurred immediately before the contrast marker is produced, whether it is the work of complaining, complimenting, endorsing, etc. The opposite side of the coin is also true: when the contrast marker appears in terminal position, it projects a much milder form of backpedaling than it would had it appeared at the beginning of the second part of the contrastive pair.[7]

Applying this line of reasoning to the hotel review example presented above, we conclude that the *compliment* is more strongly negated than the *negative endorsement*. This is so by virtue of the fact that the second pair part of the compliment—that is, the attenuation—begins with a contrast marker, whereas the second pair part of the negative endorsement—that is, the concession—ends, rather than begins, with the contrast marker. In addition, an ellipsis of three dots (…), which occurs immediately after the contrast marker found in the negative endorsement/concession contrastive pair ("So all in all, not great first impressions. Elevators are very quick *though* …"), further weakens the strength of the rebuttal, in that ellipses, as discussed earlier, convey unfinished thoughts; in this case the ellipsis indicates the somewhat noncommittal, uncertain posture of the reviewer, making his backpedaling even less definite. Accordingly, the presence of such sequentially-implicative units in this hotel review are not surprising, given that this reviewer is angry and upset—a fact amply demonstrated by the reviewer's expansive narrative complaint about his quarrel with the cab driver who complained about his tip, while the bellhops were too busy talking with other cab drivers and passersby to intervene or help with the luggage. But rather than depend on extracting relevant sentences from the review and from the review headline, which would have certainly brought out the reviewer's strongly negative sentiment, a good mining program must examine the sequence package arrangements in which useful hotel features (such as good location, comfortable bed, and very quick elevators) have been embedded and hidden in negative reviews. The alternation of compliments and attenuations or, in contrast, negative endorsements and concessions—which compose the sequence package parsing structures that have emerged indigenously in this review—might be a good place to begin.

[7] In Neustein (1981), the formal properties of cross-examination were closely analyzed for, among other things, the placement of summary contrastive facts (referred to as "contrast formulations"). The author found that the projective force of an examiner's question on the witness's next turn was occasioned by the placement of the contrast marker. It followed that stronger contrast formulations, marked by contrast markers in initial rather than terminal position, occasioned a denial from the witness, as opposed to an admission or partial concession to the attorney's accusation.

Conclusion

Review-summarization programs that mine user-generated content for opinion-related information may benefit from a close analysis of the sequence-package design of online postings. The reason is that online reviewers, like other social interactants engaged in the situated production of social order, build their reviews in situ. In so doing, they demonstrate in their blog postings the situated achievement of social order within the virtual community of online reviewers, which entails, in part, their continual negotiation of their status, role and placement within that community. Since this process is dynamic, rather than fixed, SPA offers a new natural-language understanding method which identifies the hidden attributes of reviews (attributes that, though hidden, are valuable to mobile users) by means of sequence package parsing structures that emerge indigenously as features of the *locally* achieved order of natural-language website postings. What is more, SPA's domain-independence (as well as its language-independence) render it suitable for broad application to user-generated content, not only to consumer reviews. This chapter introduces SPA as an innovative natural-language understanding method that can assist human translators in building a corpus of annotated-training data, and can eventually assist in the replacement of human annotators (supervised learning) by SPA-designed machine learning.

References

Atkinson JM, Heritage J (1984) Transcription notation. In: Atkinson JM, Heritage J (eds) Structures of social action: studies in conversation analysis. Cambridge University Press, Cambridge, pp ix–xvi

Blair CD, Keenan RL (2009) Voice interaction analysis module. Verint Americas, Mellville, NY, US Patent RE40, 634 E, 10 Feb 2009. Patent and Trade Office, Washington, DC

Button G (1990) Going up a blind alley: conflating conversation analysis and computational modeling. In: Luff P, Gilbert N, Frohlich DM (eds) Computers and conversation. Academic, London, pp 67–90

Button G, Sharrock W (1995) On simulacrums of conversation: toward a clarification of the relevance of conversation analysis for human-computer interaction. In: Thomas PJ (ed) The social and interactional dimensions of human-computer interfaces. Cambridge University Press, Cambridge, pp 107–125

Button G, Coulter J, Lee JRE, Sharrock W (1995) Computers, minds and conduct. Polity Press, Cambridge

Dahl DA (2013) Natural language processing: past, present and future. In: Neustein A, Markowitz JA (eds) Mobile speech and advanced natural language solutions. Springer, Heidelberg, pp 49–73

Denecke K (2008) Accessing medical experiences and information. In: Proceedings of workshop on mining social data (MSoDa 08) in conjunction with 18th European conference on artificial intelligence (ECAI 2008), Patras, 21–25 July 2008

Di Fabbrizio G, Stent A, Gaizauskas R (2013) Summarizing opinion-related Information for mobile devices. In: Neustein A, Markowitz JA (eds) Mobile speech and advanced natural language solutions. Springer, Heidelberg, pp 289–317

Drew P, Holt E (1988) Complainable matters: the use of idiomatic expressions in making complaints. Soc Probl 35(4):398–417

Drew P, Holt E (1998) Figures of speech: figurative expressions and the management of topic transition in conversation. Lang Soc 27(4):495–522

Gallino JA (2008) Software for statistical analysis of speech. Call Miner, Fort Myers, Florida, US Patent 7,346,509, B2, 18,Mar 2008. Patent and Trade Office, Washington, DC
Gilbert GN, Wooffitt RC, Frazer N (1990) Organizing computer talk. In: Luff P, Gilbert N, Frohlich DM (eds) Computers and conversation. Academic, London, pp 235–257
Heritage JC, Watson DR (1979) Formulations as conversational objects. In: Psathas G (ed) Everyday language: studies in ethnomethodology. Irvington Publishers, New York, pp 123–162
Hirst G (1991) Does conversation analysis have a role in computational linguistics? Comput Linguist 17(2):211–227
Hutchby I, Wooffitt R (1998) Conversation analysis: principles, practices and applications. Polity Press, Cambridge
Moore RJ (2013) A name is worth a thousand pictures: referential practice in human interactions with internet search engines. In: Neustein A, Markowitz JA (eds) Mobile speech and advanced natural language solutions. Springer, Heidelberg, pp 259–286
Moore RJ, Churchill EF, Kantamneni RGP (2011) Three sequential positions of query repair in interactions with internet search engines. In: Proceedings of ACM conference on computer supported cooperative work (CSCW11), Hangzhou, 19–23 Mar 2011
Neustein A (1981) Courtroom examination: an analysis of its formal properties. Unpublished doctoral dissertation, Department of Sociology, Boston University
Neustein A (2001) Using sequence package analysis to improve natural language understanding. Int J Speech Technol 4(1):31–44
Neustein A (2004) Sequence package analysis: a new natural language understanding method for performing data mining of help-line calls and doctor-patient interviews. In: Sharp B (ed) Proceedings of the 1st international workshop on natural language understanding and cognitive science, NLUCS 2004 in conjunction with ICEIS 2004, Porto, 13 Apr 2004, pp 64–74
Neustein A (2004) Sequence package analysis: a new global standard for processing natural language input? Globalization Insider XIII (1, 2), 18 Feb 2004, pp 1–3
Neustein A (2006a) Using sequence package analysis as a new natural language understanding method for mining government recordings of terror suspects. In: Sharp B (ed) Proceedings of the 3rd international workshop on natural language understanding and cognitive science, NLUCS 2006 in conjunction with ICEIS 2006 Cyprus, Paphos, 23–24 May 2006, pp 101–108
Neustein A (2006b) Sequence package analysis: a new natural language understanding method for improving human response in critical systems. Int J Speech Technol 9(3–4):109–120
Neustein A (2007a) Sequence package analysis: a new natural language understanding method for intelligent mining of recordings of doctor-patient interviews and health-related blogs. In: Latifi S (ed) Proceedings of the fourth international conference on information technology: new generations, ITNG 07, Las Vegas, Nevada, 2–4 Apr 2007. IEEE Computer Society, pp 441–448
Neustein A (2007b) Sequence package analysis: a new method for intelligent mining of patient dialog, blogs and help-line calls. Journal of Computers 2(10):45–51
Neustein A (2011) Sequence package analysis and soft computing: introducing a new hybrid method to adjust to the fluid and dynamic nature of human speech. In: Corchado E, Snasel V, Sedano J, Hassanien AE, Calvo JL, Slezak D (eds) Soft computing models in industrial and environmental applications. Sixth international conference SOCO 2011: advances in intelligent and soft computing, vol 87. Springer, Berlin/Heidelberg/New York, pp 1–10
Neustein A (2012) Think before you talk: the role of cognitive science in natural language processing. In: Sharp B (ed) Proceeding of NLPCS 2012, 9th international workshop on natural language processing and cognitive science in conjunction with ICEIS 2012, 28 June–1 July 2012, pp 3–11
Paprzycki M, Abraham A, Guo R (2004) Data mining approach for analyzing call center performance. In: Orchard R, Chunsheng Y, Ali M (eds) Proceedings of the 17th international conference on industrial and engineering applications of artificial intelligence and expert systems, innovations in applied intelligence. Lecture notes in computer science, vol 3029, Springer, Heidelberg, pp 1092–1101
Pieraccini R (2012) The voice in the machine. MIT Press, Cambridge, MA
Pomerantz A (1984) Pursuing a response. In: Atkinson JM, Heritage J (eds) Structure of social action: studies in conversation analysis. Cambridge University Press, Cambridge, pp 152–163

Pomerantz A (1986) Extreme case formulations: a way of legitimizing claims. Hum Stud 9(2/3):219–229

Sacks H, Schegloff EA (1979) Two preferences in the organization of reference to persons in conversation and their interaction. In: Psathas G (ed) Everyday language: studies in ethnomethodology. Irvington Publishers, New York, pp 15–21

Schegloff EA (1992) To searle on conversation: a note in return. In: Searle JR, Parret H, Verschueven J (eds) Pragmatics and beyond new series, vol 21. John Benjamins Publishing Co., Amsterdam/Philadelphia, pp 113–128

Smadja FA (1991) From N-grams to collocations: an evaluation of xtract. In: Appelt DE (ed) Proceedings of the 29th annual meeting on Association for Computational Linguistics (ACL 91), Berkeley, 18–21 June 1991. Association for Computational Linguistics, pp 279–284

Suchman L (1987) Plans and situated actions: the problem of human-machine communication. Cambridge University Press, Cambridge

Torode B (1995) Negotiating 'advice' in a call to a consumer help-line. In: Firth A (ed) The discourse of negotiation: studies of language in the workplace. Pergamon Press, Oxford, pp 345–372

Chapter 6
Getting Past the Language Gap: Innovations in Machine Translation

Rodolfo Delmonte

Abstract In this chapter, we will be reviewing state of the art machine translation systems, and will discuss innovative methods for machine translation, highlighting the most promising techniques and applications. Machine translation (MT) has benefited from a revitalization in the last 10 years or so, after a period of relatively slow activity. In 2005 the field received a jumpstart when a powerful complete experimental package for building MT systems from scratch became freely available as a result of the unified efforts of the MOSES international consortium. Around the same time, hierarchical methods had been introduced by Chinese researchers, which allowed the introduction and use of syntactic information in translation modeling. Furthermore, the advances in the related field of computational linguistics, making off-the-shelf taggers and parsers readily available, helped give MT an additional boost. Yet there is still more progress to be made. For example, MT will be enhanced greatly when both syntax and semantics are on board: this still presents a major challenge though many advanced research groups are currently pursuing ways to meet this challenge head-on. The next generation of MT will consist of a collection of hybrid systems. It also augurs well for the mobile environment, as we look forward to more advanced and improved technologies that enable the working of Speech-To-Speech machine translation on hand-held devices, i.e. speech recognition and speech synthesis. We review all of these developments and point out in the final section some of the most promising research avenues for the future of MT.

R. Delmonte, Ph.D. (✉)
Department of Linguistic Studies and Comparative Cultures, Ca' Foscari University, Dorsoduro 1075, Venezia 30123, Italy
e-mail: delmont@unive.it; project.cgm.unive.it

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_6,
© Springer Science+Business Media New York 2013

Introduction

In 2005b John Hutchins wrote the following gloomy assessment of Machine Translation (MT):

> Machine translation (MT) is still better known for its failures than for its successes. It continues to labour under misconceptions and prejudices from the ALPAC report of more than thirty years ago, and now it has to contend with widespread misunderstanding and ridicule from users of online MT services. The goal of developing fully automatic general-purpose systems capable of near-human translation quality has been long abandoned. The aim is now to produce aids and tools for professional and non-professional translation which exploit the potentials of computers to support human skills and intelligence, or which provide rough translations for users to extract the essential information from texts in foreign languages. JH (ibid., 1–5)

Since then the field of Machine Translation (MT) has dramatically changed. And in the past 3 years, the field of MT has become so huge that there is no chance of sufficiently reviewing the whole spectrum of activities, tools and resources related to the field. Therefore, I will restrict this discussion to the leading and most promising approaches.[1] The first section will be devoted to examining in detail what Statistical Machine Translation (SMT) can offer in terms of improvements to the state of the art. Then, I will dedicate section "Hybrid and Rule-Based MT Systems" to hybrid methods and systems. In section "Syntax Based Approaches: From Hierarchical to SBSMT", I will delve into syntactically based MT systems. Then section "Knowledge-Based MT Systems" will introduce knowledge, semantically-based systems. Section "Evaluation Methods and Tools" will comprise an overview of evaluation methods and section "MT for the Future" will briefly give an overview of what in my opinion may constitute the future of MT systems. This section will comprise a subsection dedicated to Speech-To-Speech MT; in another subsection promising national projects will also be reviewed. This followed by the last section in which I draw conclusions.

As JH noted, SMT research now dominates MT research. In spite of that, the great majority of commercial systems are Rule Based MT (RBMT) systems. Also most if not all professional translators are not using any of the research products (pp. 1–5). SMT systems that have reached public operational status are still only few in number, and perhaps "LanguageWeaver" – an offshoot of the research group at the University of Southern California – can be regarded as the best system offering translation systems for Arabic, Chinese, and most European languages to and from English. Always quoting from JH (ibid., 5–7):

> … there is great and increasing usage of web-based MT services (many free), such as the well-known 'Babelfish' available on Yahoo. Others include FreeTranslation, Google Translator, Bing Translator, Tarjim, WorldLingo.

[1] Consequently, I will not be concerned with commercial versions of MT systems, nor to the field of computer-assisted Translation systems or translation aids: they are all listed at http://www.hutchinsweb.me.uk/Compendium.htm, a document produced almost on a yearly basis and compiled by John Hutchins, who is also responsible for the main source of information on MT which is regularly posted on the Machine Translation Archive (http://www.mt-archive.info).

> … there are three systems specifically for translating patents: the PaTrans and SpaTrans systems developed for LingTech A/S to translate English patents into Danish …
>
> Online services are now predominantly SMT-based, e.g. 'Google Translate', 'Bing Translate' (previously 'Windows Live Translator'), 'Babelfish' (now on the Yahoo site).
>
> Probably the most significant development in MT research in Europe is the establishment of the Euromatrix project (based at Edinburgh University). Its aim is the development of open-source MT technologies applicable to all language pairs within Europe, based on hybrid designs combining statistical and rule-based methods. Perhaps best known is the Apertium framework, used for systems for Spanish, Catalan, Portuguese and Basque.

As JH noted, and I also believe, hybridization is the most interesting development of MT, and a section below will be devoted to careful examination of hybrid systems and methods. RBMT systems have been combined with SMT, and multiple subsystems are used in conjunction, such as morphological analysers, dependency parsers, and semantic engines in combination with Phrase-Based MT (PBMT). On the other side, hybrid systems that take advantage of examples have come to be used thanks to the availability of big parallel corpora of examples or translation memories, such as DGT-TM, a translation memory (sentences and their manually produced translations) organized by Ralf Steinberger from JRC and taken from the corpus Acquis Communautaire, freely downloadable at http://langtech.jrc.ec.europa.eu/DGT-TM.html. It contains all 231 language pairs from the European 22 languages, for a total of about three million sentences for most languages, 57 million in total.

Languages covered by MT have now dramatically increased, covering all European language pairs. But also Arabic and East Asian languages have become commonly translatable by MT tools, including Korean, Chinese, Japanese, Tahi, Urdu, Vietnamese, Bengali, Punjabi, Hindi, etc. Of particular interest to the US government bodies are languages like Pashto and Farsi which have also been object of translation engines.

The multi-engine approach involves the translation of a given text by two or more different MT architectures (SMT and RBMT, for example) and the integration of outputs for the selection of the 'best' output – for which statistical techniques can be used. The idea is attractive and quality improvements have been achieved, but it is difficult to see this approach as a feasible economical method for large-scale or commercial MT. An example of appending statistical techniques to rule-based MT is the experiment (by a number of researchers in Spain, Japan, and Canada) of 'statistical post-editing'. In essence, the method involves the submission (for correction and improvement) of the output of an RBMT system to a 'language model' of the kind found in SMT systems. One advantage of the approach is that the deficiencies of RBMT for less-resourced languages may be overcome. There will be more discussion on this topic below.

Statistical MT: Strength and Weaknesses

In SMT the task of translating one sentence from a source into a target language is transformed into the task of finding the "best" translation with minimum error rate: this is technically also called the minimum loss decision. In order to compose a

complete sentence, machine translation systems score small units of translation and select the fragments that, when combined together, yield the best score according to their model. The basic components of a SMT are three: a translation model, a language model and a decoder. Phrase-based SMT works as follows: source input is segmented in phrases (any sequence of words); each source phrase is automatically aligned to a target phrase on the basis of word alignment; and, eventually phrases are reordered. The decoder is responsible for the choice of best translation at sentence level: it builds translation monotonically from left to right, and the other way around for languages like Arabic and Chinese. It collects all phrase pairs that are consistent with word alignment and finds the best candidate phrase. Then it adds it to the end of partial translation, at the same time it marks the source phrase as translated. At the end of the decoding process there may be reordering. Phrase translation is the core process. There are many possible ways of segmenting and translating phrases: this is done on a probabilistic basis, and the probability distribution of the collected phrase pairs is usually based on their relative frequency. The task could be then rephrased as finding the best translation candidate hypothesis that covers all words/phrases in a sentence. Weak hypotheses are discarded and the best path is the one with best candidates. At each step the algorithm estimates costs to translate remaining part of input, and tries to find the cheapest sequence of translation option for each adjacent span of text.

Statistical MT research has explored the use of simple phrases (Och and Ney 2004), Hiero grammars (Chiang 2005), and complex S-CFG rules (Zollmann and Venugopal 2006). These more specialized translation units can more accurately describe the translation process, but they are also less likely to occur in the corpus. The increased data sparsity makes it difficult to estimate the standard SMT features which are typically computed as relative frequencies. Current weaknesses and permanent flaws are:

- Wrong word choices
- Presence of unknown words or OOVWs (Out Of Vocabulary Words)
- Mangled grammar
- Difficulties in treatment of function words (locally adding, dropping, changing)
- Lack of syntactic transformations for long-distance dependencies which require some reordering
- Lack of translation consistency (as argued by Xiao et al. 2011)
- Lack of resiliency in presence of morphologically rich languages (Chinese and English are better suited just because they are morphologically poor)
- Lack of sufficient contextual information both in translation model and in language model (trigrams are insufficient to model language discontinuities – however Chinese-English STM use a 5-gram language modeling, Shujie Liu et al. 2011), but see below.

In addition to phrase translation models, also word translation models are used, based on lexical level translation in conjunction with PBSTM. Word-based MT has a number of deficiencies that have been considered when moving to phrase-based MT, which, however, are worth while commenting.

– IBM models used have the possibility of matching one source word to many output target word – this is called fertility of the model – but not the opposite;
– Word-based models miss the majority of collocations and multiword expressions
– For many languages the syntactic structure is not symmetric and requires reordering: however word-based models penalize any such reordering and are not capable of enforcing the positioning of words at totally different places in the sentence – say the verb in Chinese and Japanese at the end of the sentence or in Arabic at the beginning compared to SVO languages.

In phrase-based models, on the contrary, the more data are available, the longer the phrases are learned, and in some cases whole sentences can be learned. Thus local context can be taken into consideration fully – the only problems may come from syntactic discontinuities and long distance dependencies, as indicated above.

Alignment of phrases goes in both directions and in this case allows for optimized results – always with IBM3 model. After aligning in both directions the results are merged and the best union or intersection is kept. As said above, phrase alignment must be consistent with word alignment and cover all words in the sentence. In this way, phrases are induced from words level alignments. Probabilities for phrases are just relative probabilities associated with each word in the phrase – a summation or a multiplication of them:

– Probability(Source/Target-phrase)=count(Source/Target-phrase)/count(Target-phrase)

The Problem of Word Alignment

Alignments are produced on the basis of IBM models 1–5, which I briefly review here (but see also Koehn 2010). Model 1 assumes that given a certain sentence length, the possible connections of words from Source to Target are all equally possible – in this way their actual order has no impact. Model 2 introduces probability to the connection between words in S/T, and it is based on position and length of the string. Models 3, considers the number of possible connections from S to T in a many-to-one fashion – thus allowing missing words or fertility. This is further conditioned in Model 4 by the identity of the words to connect in S/T. Model 5 is used to fix deficiencies. So eventually, Model 1 does lexical translation, Model 2 adds some reordering model to the output of Model 1. Then in Model 3 a fertility model is added and in Model 4 a relative reordering model, or distortion model is created always on a probabilistic basis. Distortion models are necessary every time one-to-one or monotonic alignment is insufficient, and usually ensues from many-to-many mappings (see Tiedemann 2011). The many-to-one fertility translation model is exemplified best with reference to an English to German translation system, where compound German words have to be aligned to many English words. But in some cases, English may have phrasal expressions instead of a single word in German,

so the opposite is needed. So more generally, many-to-many alignments are needed and this can be done by using GIZA++ (Och and Ney 2004) bidirectionally. The translation model is computed on the basis of word alignment, and this is regarded a critical component in SMT. However word alignment is automatically induced from parallel data, and this is usually what may constitute a real bottleneck, at least as happens in not related language pairs, where accuracy is below 80% (Hermjakob 2009). In order to produce a complete word alignment at sentence level, the system passes through the text for up to 20 iterations, to find frequent co-occurrences of words, that is words occurring in the same position in both source and target text. This usually happens at the beginning with most frequent words, that is function words – articles, prepositions, conjunctions, etc. – less frequently for content words which are more sparse. Thus eventually probabilities for content words may easily go up to 1.0, if suppose a word like "book" cooccurs with article "the" all the time. The system will look for alignments in adjacency of an already aligned word in case of misalignments: some words may come before or after another word depending on language grammar. Typically this applies to adjectives in English and Italian for instance, where English has the majority before the head noun and Italian after the head noun. Phrases will typically cover all local linguistically related positional differences in word order: decoding or translating is done monotonically once phrase alignment is terminated. Different types of constraints are applied to alignment processes as regards for instance the maximum size of segments involved in the mapping; or the maximum distance allowed for aligned segments with respect to their position in a distortion model. Translation modeling as presented above, comprises three steps: at first, sentence pairs in training corpus are aligned at word level. Then, translation pairs are extracted using some heuristic method. Lastly, maximum-likelihood estimation (MLE) is used to compute translation probabilities. The most relevant shortcoming of this method is possible inconsistent format of translation knowledge: word alignment in training versus translation pairs (phrase pairs) in decoding; then the training process which is not oriented towards translation evaluation metric (BLEU not being considered in the scoring of translation pairs – but only error minimization procedures). In this way, it is not possible to know whether translation phrases are extracted from a highly probable phrase alignment or from an unlikely one. In fact the incorrect phrases induced from inaccurate word-aligned parallel data is one of the major reasons for translation errors in phrase-based SMT systems. In Fig. 6.1 below I show the pipeline of a generic SMT using Moses (Moran et al. 2007).

Learning Improves Performance

Learning has been applied to the final decoding phase by introducing weights associated with translations and a final phase in which automatic evaluation is applied to the output of the system. This has improved dramatically the performance of SMT (see Saers et al. 2010; Saers 2011). Learning in this case is just finding model weights that make the correct translations score the best: to this aim procedures and techniques are directed to creating an optimizer, as discussed below (but see also Ambati et al. 2011).

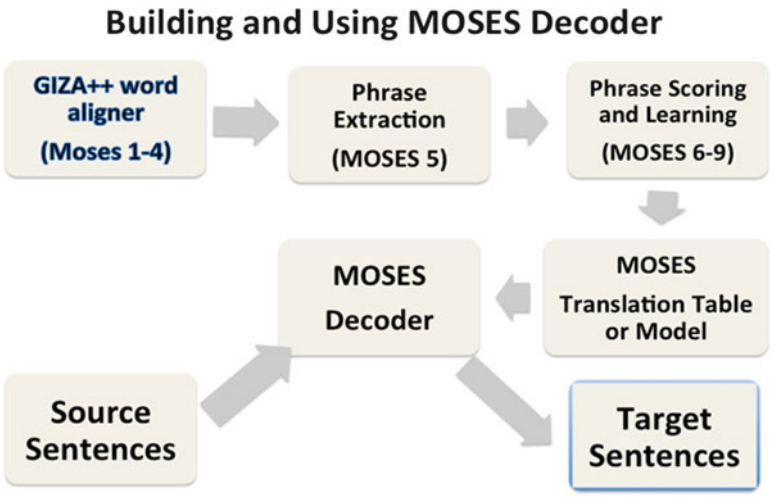

Fig. 6.1 A generic typical pipeline for an SMT system using MOSES

Discriminative models have been introduced so that translations are ranked and learned automatically by the use of features. A model consists of a number of features (e.g. the language model score). Each feature has a weight by measuring its value for judging a translation as correct. Then feature weights are optimized on training data, so that the system output matches correct translations as closely as possible. Feature weights can be adjusted and the process iterated a number of times – typically 20 iterations. Learning weights is done in a loop where the decoder generates the n-best list of candidate translation pairs. These are scored by an automatic evaluation tool – typically BLEU – then a reranking takes place which allows the system to learn best features that qualify best translations. This allows the system to change feature weights. Searching for the optimal parameters in linear models (Och and Ney 2002) of SMT has been a major challenge to the MT community. Statistical methods try to improve translation quality by minimizing error rate, and the most widely used approach to-date is Minimum-Error-Rate Training (MERT:Och 2003), which tries to find the parameters that optimize the translation quality of the one-best translation candidate, using the N-best list as an approximation of the decoder's search space. In this way, the system tries to find the best parameter that optimizes the translation quality of the first best translation candidate, and reranking follows. Reranking is done on the basis of MERT, however this method is unstable. As Cettolo et al. (2011) observe, in the last years, many efforts have been devoted to making the decoding procedure or its results more reliable. Recently, a deep investigation of the optimizer instability has been presented by Clark et al. (2011). Statistical machine translation (SMT) systems are based on a log-linear interpolation of models. Interpolation weights are typically computed by means of iterative procedures which aim at optimizing the scores of a given function. Unfortunately, as Cettolo et al. (2011) note, such a function is definitely non-convex; hence, only local optima can be reached. Moreover, it has been observed that the commonly used optimization

procedure, the N-best MERT is quite unstable. Now the focus is on improvement by reducing error rate as measured by evaluation methods. As Phillips (2011) comments, in spite of its usefulness and high adoption, MERT suffers from shortcomings of which the MT community is becoming aware. On the one hand, MERT is not designed for models with rich features and therefore leads to translations of unstable quality in such scenarios. The fluctuation in quality can even be statistically perceivable when the number of features is larger than 25 or 30 in practice. On the other hand, Smith (2006) finds that, MERT relies heavily on the behavior of parameters on the error surface, which is likely to be affected by random variances in the N-best list, and also lead to less generalizable results especially when the development set and the test set are not from exactly the same domain. As Phillips (2011) remarks, a significant challenge in building data-driven MT systems is identifying the right level of abstraction–to model translation units that both adequately reflect the data and can be estimated well.

System combination is another technique to rank the best translation, which has been applied extensively to SMT. One research line takes n-best translations of single systems, and produces the final output by means of either sentence-level combination, i.e. a direct selection from original outputs of single SMT systems (Sim et al. 2007, Hildebrand and Vogel 2008), or phrase- or word-level combination, i.e. the synthesis of a (possibly) new output joining portions of the original outputs (Rosti et al. 2007a, b, 2008; Ayan et al. 2008; He et al. 2008). These works focus on the combination of multiple machine translation systems based on different models and paradigms. More on these proposals in the section below.

Translation Models and the Problem of Overfitting

It is possible to distinguish between generative translation models (essentially, the IBM models), and the other half to various discriminative models. The first type of models induce a full probability distribution including both target and observable data and work in an unsupervised manner. The second type, on the contrary, work on labeled training data, thus supervised or semi supervised and suffer from usual related problems like data sparsity (see Tiedemann 2011:pp. 17–18). Ravi and Knight (2010) after experimenting with GIZA (sub-optimal hill-climbing) Viterbi alignment and comparing it to optimal version cast in integer linear programming (ILP), have determined that GIZA++ makes few search errors (between 5% and 15%), despite the heuristic nature of the algorithm, and that these search errors do not materially affect overall alignment (F-measure) accuracy, seen that Chinese-English averages 57–65%, and Arabic-English at 43–55% – with best values for the version that has English as target. Now, as indicated above, words that occur in totally different sentence positions, or function words that don't occur in some languages may result in poor word/phrase alignment. This problem is discussed particularly in Ulf Hermjakob's (2009), where he suggests the use of linguistic knowledge, in particular syntactic parse trees and the use of gazetteers for named-entity recognition and amalgamation. More on this topic below.

The problem of phrase-pair creation based on word alignment is nicely approached in the paper by Hyoung-Gyu Lee et al. (2010). The authors take into consideration the "collocation"-like ability of adjacent words to appear in a phrase. Different phrase segmentation will generate different translation results. To prevent bad phrases from being assigned high probabilities both collocation properties of words and multiword related probabilities taken from a corpus may be important to use. The authors characterize conditions of good segmentation which is necessary to produce high quality translation. The segmentation model they propose will consider lexical cohesion of adjacent words and the translational diversity of a word sequence as characteristics of good segmentation. To associate probabilities to such notions, they use collocation statistics from a corpus and translational entropy measures (see also Liu et al. 2010). The second is exemplified as follows: a phrase that has high translational entropy at word level but whose translational diversity at phrase level is low, should not be segmented. Though individual words in a phrase may be diversely translated with a high number of different translation pairs, the phrase may be translated with only few expressions. This would be typical of idiomatic expressions, and their model will score them very high.

Translation models should have a double function. They should well represent the training data – no gap and no bad translation; at the same time they should be able to generalize to unseen datasets.

Shujie Liu et al. (2011) discuss the problem of overfitting and note that during the training phase, the possibility of overfitting is always present. Consequently, this will hamper generalizing to unseen data: training should always be accompanied by a test phase with different datasets from the training ones. But this may not be sufficient. In fact, as the authors comment, the training phase is itself questionable because it usually optimizes on the feature weights associated to each sub-model (translation, fertility, distortion, etc.) rather than on the phrase-based translation model. At the same time, PBMT creates the phrase-based translation pairs on the basis of word alignment and the probabilities assigned by maximum-likelihood estimation (MLE). The paper proposes a new unified framework to add a discriminative translation sub-model into the conventional linear framework (more on discriminative models below), and the sub-model is optimized with the same criterion as the translation output is evaluated (BLEU in our case). In this case, each translation pair is a feature in itself, and the training method can affect the pairs directly so as to handle overfitting (ibid., 181).

As the authors clearly demonstrate, the translation model will overestimate probabilities for long translation pairs and underestimate those for short phrases. This will cause overfitting and will prevent the system to generalize to unseen data where those short phrases may appear. Filtering away long phrases is also not the best solution, because they may be useful for good translations and cannot be done away with in case they contain non compositional semantic material like idiomatic phrases. Wuebker et al. (2010) used the approach called leaving-one-out (L1O) to deal with overfitting and forced alignment to deal with the errors introduced by incorrect word alignment. The basic idea is to use the trained SMT decoder to re-decode the training data, and then use the decoded result to re-calculate translation

pair probabilities. Since the correct target sentence (i.e. the target side of training data) is not guaranteed to be generated by SMT decoder, forced alignment is used to generate the correct target sentence by discarding all phrase translation candidates which do not match any sequence in the correct target sentence. Since only the correct target sentence can be generated, language model is useless during decoding, so the weight for language model is set to be zero.

Scalable training methods (Perceptron, MIRA and OWL-QN) are used to train the purely discriminative translation model with a large number of features. In order to optimize SMT performance, scalable training tunes the weights to push the best translation candidate upward to be the first one in n-best list. In order to perform scalable training, the n-best candidates should be ranked according to the similarity with the correct target sentence. BLEU is the most natural choice for a similarity measure as it is also the ultimate evaluation criterion. However, BLEU is a document-level metric rather than sentence-level.

Modern phrasal SMT systems (such as Koehn et al. 2003) derive much of their power from being able to memorize and use long phrases. Phrases allow for non-compositional translation, local reordering and contextual lexical choice. However phrases are fully lexicalized, which means they generalize poorly to even slightly out of-domain text. In an open competition (Koehn and Monz 2006) systems trained on parliamentary proceedings were tested on text from 'news commentary' web sites, a very slightly different domain. The nine phrasal systems in the English to Spanish track suffered an absolute drop in BLEU score of between 4.4% and 6.34% (14–27% relative). The treelet system of Menezes et al. (2006) fared somewhat better but still suffered an absolute drop of 3.61%. Clearly there is a need for approaches with greater powers of generalization. There are multiple facets to this issue, including handling of unknown words, new senses of known words etc. At the end of the chapter I will return to the topic of language and translation modeling.

Hybrid and Rule-Based MT Systems

Statistical machine translation (SMT) (Koehn 2010) is currently the leading paradigm in machine translation (MT) research. SMT systems are very attractive because they may be built with little human effort when enough monolingual and bilingual corpora are available. However, bilingual corpora are not always easy to harvest, and they may not even exist for some language pairs. On the contrary, rule-based machine translation systems (RBMT) (Hutchins and Somers 1992) may be built without any parallel corpus; however, they need an explicit representation of linguistic information, whose coding by human experts requires a considerable amount of time.

Rule-Based MT or RBMT for short are by far the mostly used commercial systems still today. This might change in the future. However, it is a fact that SMT has not yet been able to supersede previous work being done on MT which was mainly done in a rule-based fashion.

From a general perspective, hybrid systems are certainly the winning solution. Here I am referring to systems that mix up in a perspicuous manner statistical and non statistical approaches. This can be done in many ways, here are some reported in the literature:

- Using translation memories together with domain trained statistical translation models – this can be done better by using example-based techniques and resources
- Using statistical post-editing before manual supervision with domain trained translation models
- In lack of domain bitexts, providing a dictionary of translation pairs
- Using morphological decomposition for morphologically rich languages (Arabic, German, Italian, French …)
- Using multiword preprocessing in both parallel texts before running language models to reduce semantic uncertainty

When both parallel corpora and linguistic information exist, a hybrid approach may be taken in order to make the most of such resources. In Thurmair (2009) a new hybrid approach is presented which enriches a phrase-based SMT system with resources taken from shallow-transfer RBMT. Shallow-transfer RBMT systems do not perform a complete syntactic analysis of the input sentences, but they rather work with much simpler intermediate representations. Hybridisation between shallow-transfer RBMT and SMT has not yet been explored. Existing hybridisation strategies usually involve more complex RBMT systems and treat them as black boxes, whereas the approach improves SMT by explicitly using the RBMT linguistic resources. They provide an exhaustive evaluation of their hybridisation approach and of the most similar one (Eisele et al. 2008), on the Spanish–English and English–Spanish language pairs by using different training corpus sizes and evaluation corpora.

Rule-based machine translation systems heavily depend on explicit linguistic data such as monolingual dictionaries, bilingual dictionaries, grammars, and structural transfer rules (Hutchins and Somers 1992). Although some automatic acquisition is possible (see Caseli et al. 2006), collecting these data usually requires the intervention of domain experts (mainly, linguists) who master all the encoding and format details of the particular MT system. It could be interesting, however, to open the door to a broader group of non-expert users who could collaboratively enrich MT systems through the web.

Esplà-Gomis et al. (2011) focus on these kinds of dictionaries, which basically have two types of data: paradigms (that group regularities in inflection) and word entries. The paradigm assigned to many common English verbs, for instance, indicates that by adding the ending -ing, the gerund is obtained. Paradigms make easier the management of dictionaries in two ways: by reducing the quantity of information that needs to be stored, and by simplifying revision and validation thanks to the explicit encoding of regularities in the dictionary.

Bilingual dictionaries are the most reused resource from RBMT. They have been added to SMT systems since its early days (Brown et al. 1993). One of the simplest strategies, which has already been put into practice with the Apertium bilingual dictionaries (Tyers 2009; Sanchez-Cartagena and Pérez-Ortiz 2010),

consists of adding the dictionary entries directly to the parallel corpus. In addition to the obvious increase in lexical coverage, Schwenk et al. (2009) state that the quality of the alignments obtained is also improved when the words in the bilingual dictionary appear in other sentences of the parallel corpus. However, it is not guaranteed that, following this strategy, multi-word expressions from the bilingual dictionary that appear in the SL sentences are translated as such because they may be split into smaller units by the phrase-extraction algorithm. Other approaches go beyond simply adding a dictionary to the parallel corpus. For instance, Popovic´ and Ney (2006) propose combining that strategy with the use of hand-crafted rules to reorder the SL sentences to match the structure of the TL.

Although RBMT transfer rules have also been reused in hybrid systems, they have been mostly used implicitly as part of a complete RBMT engine.

For instance, Dugast et al. (2008) show how a PBSMT system can be bootstrapped using only monolingual data and an RBMT engine; RBMT and PBSMT systems can also be combined in a serial fashion (Dugast et al. 2007). Another remarkable study (Eisele et al. 2008) presents a strategy based on the augmentation of the phrase table to include information provided by an RBMT system. In this approach, the sentences to be translated by the hybrid system are first translated with an RBMT system. Then a small phrase table is obtained from the resulting parallel corpus. Phrase pairs are extracted following the usual procedure (Koehn 2010, Sect. 5.2.3) which generates the set of all possible phrase pairs that are consistent with the word alignments. In order to obtain reliable word alignments, they are computed using an alignment model previously built from a large parallel corpus. Finally, the RBMT-generated phrase table is directly added to the original one. Another approach is to generate phrase pairs directly which match either an entry in the bilingual dictionary or a structural transfer rule, thus preventing them from being split into smaller phrase pairs even if they would be consistent with the word alignments. In this way, there is no need for a large parallel corpus from which to learn an alignment model.

España-Bonet et al. (2011) present a system which is guided by a RBMT. In their introduction, they explain why the hybridization is necessary,

> It is well known that rule-based and phrase-based statistical machine translation paradigms (RBMT and SMT, respectively) have complementary strengths and weaknesses. First, RBMT systems tend to produce syntactically better translations and deal with long distance dependencies, agreement and constituent reordering in a better way, since they perform the analysis, transfer and generation steps based on syntactic principles. On the bad side, they usually have problems with lexical selection due to a poor handling of word ambiguity. Also, in cases in which the input sentence has an unexpected syntactic structure, the parser may fail and the quality of the translation decreases dramatically.
>
> On the other side, phrase-based SMT models usually do a better job with lexical selection and general fluency, since they model lexical choice with distributional criteria and explicit probabilistic language models. However, SMT systems usually generate structurally worse translations, since they model translation more locally and have problems with long distance reordering. They also tend to produce very obvious errors, which are annoying for regular users, e.g., lack of gender and number agreement, bad punctuation, etc. Moreover, the SMT systems can experience a severe degradation of performance when applied to corpora different from those used for training (out-of-domain evaluation). (ibid., 554)

The hybrid architecture tries to get the best of both worlds: the RBMT system should perform parsing and rule-based transfer and reordering to produce a good structure for the output, while SMT helps the lexical selection by providing multiple translation suggestions for the pieces of the source language corresponding to the tree constituents. The final decoding accounts also for fluency by using language models, and can be monotonic (and so, fast) because the structure has been already decided by the RBMT component.

System combination, either serial or by a posterior combination of systems' outputs, is a first step towards hybridization. Although it has been shown to improve translation quality, the combination does not represent a real hybridization since systems do not interact among them (see Thurmair 2009) for a classification of HMT architectures. In the case of actual interdependences, one of the systems in action leads the translation process and the other ones strengthen it. Much work has been done in building systems in which the statistical component is in charge of the translation and the companion system provides complementary information. For instance, Eisele et al. (2008) and Chen and Eisele (2010) introduce lexical information coming from a rule-based translator into an SMT system, in the form of new phrase pairs for the translation table. In both cases results are positive on out-of-domain tests.

The opposite direction is less explored. In this case, the RBMT system leads the translation and the SMT system provides complementary information. Habash et al. (2009) enrich the dictionary of a RBMT system with phrases from an SMT system (see also Alkuhlani and Habash 2011). Federmann et al. (2010) use the translations obtained with a RBMT system and substitute selected noun phrases by their SMT counterparts. Globally, their results improve the individual systems when the hybrid system is applied to translate into languages with a richer morphology than the source. In Figure 6.2 below there is a pipeline for a generic Hybrid System that combines a Rule Based approach with Statistical Models.

Specific Issues in Hybrid MT

A number of specific issues are dealt with inside this framework, even though they may certainly be regarded as general problems of SMT. In particular, the treatment of English particle and of function words, is a topic that has developed into a number of interesting techniques.

Morpheme-based SMT system (SMTm) a second variant of the SMT system was used to address the rich morphology of Basque. In this system, words are split into several morphemes by using a Basque morphological analyzer/lemmatizer. The aim is to reduce the sparseness produced by the agglutinative nature of Basque and the small amount of parallel corpora. Adapting the baseline system to work at the morpheme level mainly consists of training Moses on the segmented text. The SMT system trained on segmented words will generate a sequence of morphemes. So, in order to obtain the final Basque text from the segmented output, a word-generation post-process is applied. Details on this system can be found in (Labaka 2010).

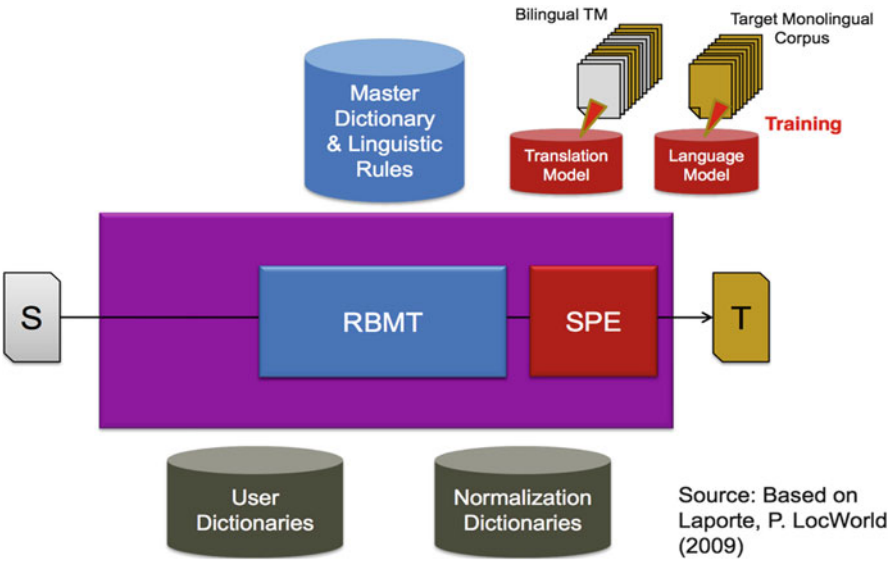

Fig. 6.2 Hybrid Systems

Matxin is another interesting hybrid rule-based system, an open-source Spanish-Basque RBMT engine (Alegria et al. 2007), following the traditional transfer model. The system consists of three main components: (1) analysis of the source sentence into a dependency tree structure; (2) transfer from the source language dependency tree to a target language dependency structure; and (3) generation of the output translation from the target dependency structure. The SMatxinT architecture is based on the three following principles: (1) generally, the final translation should be based on RBMT system's syntactic rearrangements; (2) the hybrid system must have the chance of using SMT-based local translations to improve lexical selection; and (3) it should be able to recover from potential problems encountered in the analysis, using longer SMT translations.

Phillips 2011 discusses other ways of overcoming shortages of SMT, introducing ways to incorporate the context for each translation instance. To overcome these deficiencies Phillips proposes to model each instance of translation. An instance of translation is the realization of a source and corresponding target phrase at one specific location in the corpus. He defines his method as follows:

> An instance of translation is the realization of a source and corresponding target phrase at one specific location in the corpus. We score each translation instance with a series of features that examine the alignment, context, genre, and other surroundings. Our model then combines these translation instances in a weighted summation. This approach conveniently side-steps the challenges of estimation sparsity because our model is not based on relative frequency estimates. The weighting of translation instances relates to methods for domain-adaptation of SMT models, but our implementation is fundamentally different in that we do not alter or re-weight the training data. Instead, our model directly embodies the notion that

> not all translations are equal and individually evaluates the relevance of each translation instance… Evidence for a translation unit θ will generally be present at multiple locations within the training data. The features for θ operate over this set of translation instances and are generally computed as relative frequencies. A common feature, for example, is the number of times source and target instances are aligned divided by the total occurrences of source instances.
>
> Our model for translation is fundamentally different in that our translation units are not abstract phrase pairs or grammar rules … the core component of our model is a feature function which allows the user to easily add new sources of knowledge to the system. However, our feature function ϕ evaluates one specific instance of translation instead of scoring the entire set of translation instances.

In fact, they produced this new statistical model because they wanted explicitly to incorporate ideas coming from EBMT,

> For illustration, consider that the translation instances for a given phrase pair occur in a variety of sentences within the training data. Some instances may include an inconsistent word alignment from within the selected phrase pair to a word in the remainder of the sentence. Our model allows us to learn from these translation instances, but discount them by including a feature in ϕ which measures the likelihood of the phrasal alignment given the words outside the phrase pair. This differs from the standard SMT approach where phrase alignment is a binary decision. The same principle also applies if we want to include additional non-local information such as genre or context within the model. A traditional SMT model requires new translation units conditioned on the extra information whereas our approach incorporates the extra information as features of ϕ and calculates a score over all instances.
>
> One of the motivations for this model was to combine ideas from Statistical MT and Example-Based MT. Many EBMT systems rely on heuristics and lack a well-defined model, but our per-instance modeling is generally reflective of an 'EBMT approach.'

English Particles and Function Words

Ma Jianjun et al. (2011) address the problem of correctly translating English particles (adverbs and prepositions) into Chinese. They introduce POS tags in the corpus, and thus tags become an important feature for the Maximum Entropy model. For that purpose, they use the Stanford tagger. However, in order to improve the results, they have to proceed to some post-processing operation with rules that take into account typical phrasal verb collocations from a manually built collocation bank.

In practice, many function words do not have the exact counterparts in the other language and will not align to any words (i.e. align to NULL) in the results of word alignment. Furthermore, due to the high frequencies of function words, they could be associated with any content words to form bilingual phrases which might be quite noisy.

Consequently, many target function words may be either missing or inappropriately generated in the translations. This not only degrades the readability but also impairs the quality of content word translations.

The incompleteness of target function word generation seems to be mainly caused by the noisy translation knowledge automatically learnt based on word alignment.

In particular, some words serve to express (language-specific) grammatical relations only, and thus they may have no counterpart in another language. This problem is nicely treated in Liu et al. (2011). They divide up words into two subcategories: spurious and non spurious words. The first type should be aligned to a null token. For example the Chinese words "hen" and "bi" have no counterparts on the other side, neither do the English words "does" and "to". To deal with the spurious words in sentence pairs, IBM models 3, 4 and 5 (Brown et al. 1993) introduce a special token null, which can align to a source/target word. Fraser and Marcu (2007a) proposed a new generative model called LEAF, which is similar to the IBM models, in which words are classified into three types instead of two: spurious words, head words (which are the key words of a sentence) and non-head words (modifiers of head words).

"Spurious" words usually have no counterpart in other languages, and are therefore a headache in machine translation. The authors propose a novel framework, called skeleton-enhanced translation, in which a conventional SMT decoder can boost itself by considering the skeleton of the source input and the translation of such skeleton. The skeleton of a sentence is the sentence with its spurious words removed. Two models for identifying spurious words, are introduced. The first one is a context-insensitive model, which removes all tokens of certain words. The second one is a context-sensitive model, which makes separate decision for each word token. The authors also elaborate two methods to improve a translation decoder using skeleton translation. One is skeleton-enhanced re-ranking, which re-ranks the n-best output of a conventional SMT decoder with respect to a translated skeleton. Another is skeleton-enhanced decoding, which re-ranks the translation hypotheses of not only the entire sentence but any span of the sentence. Their experiments show significant improvement (1.6 BLEU) over the state-of-the-art SMT performance.

These two methods are generative models for word alignment. Nevertheless they cannot be used directly in the conventional log-linear model of statistic machine translation (SMT). The conventional phrase-based SMT captures spurious words within the phrase pairs in the translation table. As Liu et al. (2011) note, the existence of spurious words in training data leads to a certain kind of data sparseness. For example, "na bi qian" and "na xie qian" share the same translation ("that money"). If the spurious words ("bi" and "xie") are removed, then the two entries in translation table, and the associated statistics, can be combined into one. However, while spurious words lead to the harmful effect of data sparseness, they are useful in certain aspects in translation. To cope with this problem, as automatic word alignment is far from perfect, in keeping a high precision of spurious word deletion. It is stipulated that a word token is not to be removed unless the model assigns a high probability to the deletion decision.

Correct translation of function words into Chinese is discussed in a paper by Cui et al. (2011). They have been interested in the subject because "… function words play an important role in sentence structures and express grammatical relationships with other words"(ibid., 139). Most statistical machine translation (SMT) systems do not pay enough attention to translations of function words which are noisy due to data sparseness and word alignment errors. Their method is designed to separate

the generation of target function words from target content words in SMT decoding. With this method, the target function words are deleted before the translation modeling while in SMT decoding they are inserted back into the translations. To guide the target function words insertion, a new statistical model is proposed and integrated into the log-linear model for SMT. This can lead to better reordering and partial hypotheses ranking. As shown by experimental results, their approach improves the SMT performance significantly on Chinese-English translation task.

For example, when considering the top eight function words, about 63.9% of Chinese function word occurrences are not aligned to any English words and about 74.5% of Chinese sentences contain at least one unaligned Chinese function word. On the English side, about 36.5% of English function word occurrences are not aligned to any Chinese words and about 88.8% of English sentences contain at least one unaligned English function word.

Combining Translation Memories with EBMT

Even though over the past two decades, machine translation has shown very promising results, a large number of languages exist which suffer from the scarcity of parallel corpora, e.g. Indic languages, sign languages etc. SMT approaches have yielded low translation quality for these poorly resourced languages (Khalilov et al. 2010). It is often the case that domain-specific translation is required to tackle the issue of scarce resources, but it can still suffer from very low accuracy within the SMT framework, even for homogeneous domains (Dandapat et al. 2010). Although SMT and EBMT are both data-driven approaches to MT, both of them have their own advantages and limitations. Typically, an SMT system works well with significant amounts of training data. In contrast, an EBMT approach can be developed with a limited example-base (Somers 2003); also, as with any other data-driven system, an EBMT system works well when training and test sets are quite close in nature. This is because EBMT systems reuse the segments of test sentences that can be found in the source side of the example-base at runtime (see Brown 1996). Keeping these points in mind is important in order to develop an MT system of reasonably good quality based on limited amounts of data. In this direction, they examine different EBMT approaches which can handle the problem of data sparseness. It is often the case that EBMT systems produce a good translation where SMT fails and vice versa. In order to harness the advantages of both approaches, they use a careful combination of both EBMT and SMT to improve translation accuracy.

Two alternative approaches are adopted to tackle the above problems. First there is a compiled approach to EBMT which essentially produces translation templates during the training stage, based on the description in (Cicekli and Güveniri 2001). The second attempt presents a novel way of integrating translation memory (TM) into an EBMT system. Starting with the user's TM as a training set, additional sub-sentential translation units (TUs) are extracted based on the word alignments produced by an SMT system. These sub-sentential TUs are used both for alignment and recombination after the closest matching example to the input is found in the matching stage of our EBMT system.

Simard et al. (2007) note that TM has some notable advantages over most data-driven MT systems. The most obvious is its ability to translate predictably and (near-) perfectly any input that it has seen previously. Another quality of TM is its ability to find approximate matches and to let the user adapt system behavior to his/her own tolerance to errors by fixing the similarity threshold on such matches; in other words, TM's benefit from a highly effective confidence estimation mechanism. If machine translation is to be integrated successfully in the CAT environment, it should begin by catching up with TM on these aspects. This requires two things: (1) the MT system should behave more like a TM in the presence of high-similarity matches. In practice, this can be achieved by combining the two technologies, i.e. by building a combination MT system that incorporates a TM component. And (2) just like existing TM systems, the combined MT system should provide the user with means to filter out translations that are less likely to be useful. It has sometimes been proposed (see e.g. Heyn 1996) that MT should be used within a CAT environment only when the TM fails to retrieve something useful. Unfortunately, this has the effect of relegating the MT system to the task of translating only the sentences that are most unlike previously seen ones. For data-driven systems, this means translating only the "harder" sentences and missing the chance to do a better job than the TM. The reason why MT is often treated as a last resort lies in the fact that translators tend to see its performance as unpredictable and, as a result, overly likely to waste their time.

Example-based MT has the problem of coverage and the fragments used are not decomposable so they have limited flexibility. This may be improved by Translation Memories made available by users in a specific domain, such as the just released databank of TMs from JRC (see above).

In 2003, the idea of combining knowledge coming from Translation Memories, which is very domain localised, and EBMT was not yet clearly formulated. Even the organizers, Carl and Way, of the corresponding workshop admit this. While translation memory systems are used in restricted domains, SBMT systems require training on huge, good quality bilingual corpora. As a consequence TMs can hardly be applied as a general purpose solution to MT, and SBMT as yet cannot produce complex translations to the desired quality, even if such translations are given to the system in the training phase. EBMT seeks to exploit and integrate a number of knowledge resources, such as linguistics and statistics, and symbolic and numerical techniques, for integration into one framework. In this way, rule-based morphological, syntactic and/or semantic information is combined with knowledge extracted from bilingual texts which is then re-used in the translation process.

However, it is unclear how one might combine the different knowledge resources and techniques in an optimal way. In EBMT, therefore, the question is asked: what can be learned from a bilingual corpus and what needs to be provided manually? Furthermore, it is uncertain how far the EBMT methodology can be pushed with respect to translation quality and/or translation purpose. Finally, one wonders what the implications and consequences are for the size and quality of the reference translations, (computational) complexity of the system, sizeability and transportability, if such an approach is taken.

Sanchez-Cartagena et al. (2011), extensively evaluate a new hybridisation approach. It consists of enriching the phrase table of a phrase-based statistical

machine translation system with bilingual phrase pairs matching transfer rules and dictionary entries from a shallow-transfer rule-based machine translation system. The experiments conducted show an improvement in translation quality, specially when the parallel corpus available for training is small (see also Sanchez-Martinez and Forcada 2009), or when translating out-of-domain texts that are well covered by the shallow-transfer rule-based machine translation system.

In their paper Dandapat et al. (2011) address the issue of applying example-based machine translation (EBMT) methods to overcome some of the difficulties encountered with statistical machine translation (SMT) techniques. They adopt two different EBMT approaches and present an approach to augment output quality by strategically combining both EBMT approaches with the SMT system to handle issues arising from the use of SMT. They use these approaches for English to Turkish translation using the IWSLT09 dataset. Improved evaluation scores (4% relative BLEU improvement) were achieved when EBMT was used to translate sentences for which SMT failed to produce an adequate translation.

Ebling et al. (2011) present Marclator and its ability to chunk based on the so-called "Marker Hypothesis" (Green 1979), which is a psycholinguistic hypothesis stating that every language has a closed set of elements that are used to mark certain syntactic constructions. Marclator system segments both the training and the test data into chunks, where the set of elements includes function words and bound morphemes (-ing as an indicator of English progressive-tense verbs). The interesting point is that Marclator chunking module solely considers function words as indicators of chunk boundaries, and head words are included in their inflected forms. In fact, each function word (Marker word) triggers the opening of a new chunk, provided that the preceding chunk contains at least one non-Marker word.

Chunk example: He was | on the bus

Typical problems inherent in this approach are the chunks of an input sentence that often cannot be found in the example base. So the goal is to increase the chunk coverage of a system. Gough and Way (2003) extended a precursor to Marclator by including an additional layer of abstraction: producing generalized chunks by replacing the Marker word at the beginning of a chunk with the name of its category.

For example: of a marathon | <PREP> a marathon

OPENMATREX is another system based on the marker hypothesis reported lately in Banerjee et al. 2011. As the authors comment in the conclusion, "OpenMaTrEx comprises a marker-driven chunker, a collection of chunk aligners, tools to merge (hybridise") marker-based and statistical translation tables, two engines a simple proof-of-concept monotone "example-based" recombination engine and a statistical decoder based on Moses, and support for automatic evaluation. It also contains support for "word packing" to improve alignment."(ibid. 14) The performance of the system shows improvements over the purely statistical mode.

In CMU-EBMT II, the system generalizes both the training and the test set: it recursively replaces words and phrases that are part of an equivalence class with the corresponding class tags. Syntactic classes are applied before semantic classes. In training data, a generalization is performed only if a member of a particular equivalence class is found in both the SL and the corresponding TL sentence. Eventually, in CMU-EBMT III the test set has all members of an equivalence class that are replaced recursively.

The matching process is equivalent to that of the purely lexical CMU-EBMT system, with the apparent difference that here, two matching levels: a lexical and a generalized one exist. The alignment proceeds in the same way as in CMU-EBMT. Following this, the rules that were stored during the generalization of the input sentence are applied in reverse so as to transform the generalized TL fragments into word form TL fragments. The system carries out translations by matching chunks first. If the system does not find a chunk in the example base, it proceeds to replace the Marker word at the beginning of a chunk with its corresponding Marker tag and to search for the resulting generalized chunk in the example base (if this attempt fails, the system reverts to word-by-word translation).

One major source of errors is chunk-internal boundary friction. Boundary friction is normally caused by combining two separate translation units that do not agree in grammatical case with the introduction of Marker-based templates. It can also take place within a single chunk, i.e., when a Marker word is inserted that does not agree with the grammatical properties of the rest of the chunk. In the case of translating from English to German, inserting TL Marker words in a context-insensitive manner (as is done in System 1) is error prone. Due to the morphological richness of German, an English Marker word can correspond to multiple word forms of the same lemma on the German side e.g., English Marker word "are" can correspond to German Marker words "bist, sind" and "seid". Example: for "are you sure … /sind du sicher …" where the chunk-internal boundary friction causes a combination of "sind" and "du" which is grammatically incorrect.

Eventually I want to include a short note on Graph-Based Learning approaches, which will be explained in a section below – that can be likened to a probabilistic implementation of translation memories (Maruyana and Watanabe 1992; Veale and Way 1997). Translation memories are (usually commercial) databases of segment translations extracted from a large database of translation examples. They are typically used by human translators to retrieve translation candidates for subsequences of a new input text. Matches can be exact or fuzzy; the latter is similar to the identification of graph neighborhoods in our approach. However, the GBL scheme propagates similarity scores not just from known to unknown sentences but also indirectly, via connections through other unknown sentences. Marcu 2001 reported the combination of a translation memory and statistical translation; however, this is a combination of word-based and phrase-based translation predating the current phrase-based approach to SMT.

Automatic Post-editing for Translator CAT Tools

The translation quality of MT has been improving but has not reached an adequate level compared with human translation. As such, manual evaluation and post-editing constitute an essential part of the translation processes. To make the best use of MT, human translators are urged to perform post-editing efficiently and effectively. Therefore there is a huge demand for MT to alleviate the burden of manual post-editing.

Alleviating the burden for human post-editing is the aim of research efforts in the direction of producing automatic procedures that work possibly on the basis of the output of STM or RBMT and TM or at least strongly domain limited bitexts. In fact "bitext"

is not synonymous with parallel corpora, as Tiedemann 2011 notes. Suzuki (2011), working for Toshiba, has built a quality prediction model with regression analysis for Japanese English and viceversa APE, where confidence estimation (CE) is considered as also (Specia et al. 2009a; 2009b) do, by estimating a continuous translational quality score for each sentence, using PLS (Partial Least Squares) regression analysis. Since Rule-based MT (RBMT) is generally more stable in translation quality than SMT, it can make it easier to integrate the post-editing into the translation processes. This, however, is also a weak point of RBMT because post-editors are forced to repeatedly correct the same kind of errors made by MT systems (see Roturier 2009). Statistical post-editing (SPE) techniques have been successfully applied to the output of Rule Based MT (RBMT) systems. In the computing assisted translation process with machine translation (MT), post-editing costs time and efforts on the part of human. To solve this problem, some have attempted to automate post editing. Post-editing isn't always necessary, however, when MT outputs are of adequate quality for human. This means that we need to be able to estimate the translation quality of each translated sentence to determine whether post-editing should be performed. While conventional automatic metrics such as BLEU, NIST and METEOR, require the golden standards (references), for wider applications we need to establish methods that can estimate the quality of translations without references. The paper presents a sentence-level automatic quality evaluator, composed of an SMT phrase-based automatic post-editing (APE) module and a confidence estimator characterized by PLS regression analysis. It is known that this model is a better model for predicting output variable than a normal multiple regression analysis when the multicollinearity exists between the input variables. Experiments with Japanese to English patent translations show the validity of the proposed methods.

Recognizing that SMT is better suited to correct frequent errors to appropriate expressions, some (Simard et al. 2007; Lagarda et al. 2009) have proposed to use SMT for an automatic post-editor and built an automatic post-editing module, where MT outputs are regarded as source sentences and manually post-edited/translated results as target sentences.

Béchara et al. (2011) investigate the impact of SPE on a standard Phrase-Based Statistical Machine Translation (PB-SMT) system, using PB-SMT both for the first-stage MT and the second stage SPE system. Their results show that, while a naive approach to using SPE in a PB-SMT pipeline produces no or only modest improvements, a novel combination of source context modeling and thresholding can produce statistically significant improvements of 2 BLEU points over baseline using technical translation data for French to English.

Simard et al. (2007a) train a "mono-lingual" PB-SMT system (the Portage system) on the output of an RBMT system for the source side of the training set of the PB-SMT system and the corresponding human translated reference. A complete translation pipeline consists of a rule-based first-stage system, whose output on some (unseen) test set, in turn, is translated by the second-stage "mono-lingual" SPE system. Simard et al. (2007a) present experiments using Human Resources and Social Development (HRSDC) Job Bank1 French and English parallel data. They found that in combination, the RBMT system post-edited by the PB-SMT system performed

significantly better than each of the individual systems on their own. Simard et al. (2007a) also tested the SPE technique with Portage PB-SMT both as first-stage MT and as second stage SPE system (i.e. Portage post-editing its own output) and reported that nothing could be gained. In a number of follow-up experiments, Simard et al. (2007b) used an SPE system to adapt RBMT-systems to a specific domain, once again using Portage in the SPE phase. Adding the SPE system produced BLEU score increases of about 20 points over the original RBMT baseline.

SPE was also applied in an attempt to improve Japanese to English patent translations. Teramusa (2007) uses RBMT to translate patent texts, which tend to be difficult to translate without syntactic analysis. Combining RBMT with SPE in the post-editing phase produced an improved score on the NIST evaluation compared to that of the RBMT system alone. Dugast et al. (2007) report research on combining SYSTRAN with PB-SMT systems Moses and Portage. Comparison between raw SYSTRAN output and SYSTRAN + SPE output shows significant improvements in terms of lexical choice, but almost no improvement in word order or grammaticality. Dugast et al. (2009) trained a similar post-editing system with some additional treatment to prevent the loss of entities such as dates and numbers.

Oflazer and El-Kahlout (2007) explore selective segmentation-based models for English to Turkish translation. As part of their experiments they present a short section at the end of the paper on statistical post-editing of an SMT system, which they call model iteration. They train a post-editing SMT model on the training set decoded by the first stage SMT model and iterate the approach, post-editing the output of the post-editing system. BLEU results show positive improvements, with a cumulative 0.46 increase after two model iterations. It is not clear whether the result is statistically significant. The experiments follow the statistical post-editing design of Simard et al. (2007a), where the output of a first-stage system is used to train a mono-lingual second stage system, that has the potential to correct or otherwise improve on (i.e. post-edit) the output of the first-stage system. The experiments use PB-SMT systems throughout both stages. The objective is to investigate in more detail whether and to what extent state-of-the-art PBSMT technology can be used to post-edit itself, i.e. its own output.

Blain et al. (2011) report on work on post-editing by Systran and Symantec where they define what they call a Post-Editing Action (PEA) typology on the basis of a detailed analysis of errors, which we report here below (166–167):

Noun-Phrase (NP) – related to lexical changes.

- Determiner choice – change in determiner
- Noun meaning choice – a noun, replaces another noun, changing its meaning
- Noun stylistic change – a synonym replaces a noun (no meaning change)
- Noun number change
- Case change
- Adjective choice – change in adjective choice for better fit with modified noun
- Multi-word change – multiword expression change (meaning change)
- NP structure change – structure change of NP but the sense is preserved

- Verbal-Phrase (VP) – related to grammatical changes
- Verb agreement – correction of agreement in verb
- Verb phrase structure change
- Verb meaning choice – a verb replaces another verb, changing its meaning
- Verb stylistic change – a synonym replaces a verb.

Preposition change

- Co-reference change – generally through introduction/removal of a pronoun, or change of a definite to possessive determiner
- Reordering – repositioning of a constituent at a better location (adjective, adverb)
- PE Error – Post-editor made a mistake in his review
- Misc style – unnecessary stylistic change
- Misc – all PEAs that we cannot classify

The UNL: Universal Networking Language

In a paper online, Alansary et al. present the UNL concisely and report some recent data. One of the challenging missions that the UNL system has to face is to translate the Encyclopedia of Life Support System (EOLSS) which is the largest on-line Encyclopaedia; it includes more than 120,000 web pages and it increases constantly. The translation results are reported as reaching a morphological accuracy of 90%, a syntactic accuracy of 75% and a semantic accuracy of 85%. The adopted approach in the translation in this abstract follows a different way, it translates from a semantically-based Interlingua to different human languages. The UNL (see Adly and Alansary 2009) has been introduced by the United Nations University, Tokyo, to facilitate the transfer and exchange of information over the internet. The semantic representation is an artificial language which describes the meaning of sentences in terms of the schema of semantic nets. It aims to represent all sentences that have the same meaning in all natural languages using a single semantic graph. Once this graph is built, it is possible to decode it to any other language. UNL is used not only in machine translation and other natural language processing tasks, but also in a wide variety of applications ranging from e-learning platforms to management of multilingual document bases. Working at the semantic level, the UNL is language-independent: in particular, it follows the schema of semantic nets-like structure in which nodes are word concepts and arcs are semantic relations between these concepts. In this scheme, a source language sentence is converted to the UNL form using a tool called the EnConverter. EnConverter is a language independent parser that provides synchronously a framework for morphological, syntactic and semantic analysis. Subsequently, the UNL representation is converted to the target language sentence by a tool called the DeConverter. The DeConverter is a language independent generator that provides a framework for syntactic and morphological generation as well as co occurrence-based word selection for linguistic collocations.

It can deconvert UNL expressions into a variety of native languages, using a number of linguistic data such as Word Dictionary, Grammatical Rules and Co-occurrence Dictionary of each language. UNL's main task and purpose is translating The Encyclopedia Of Life Support Systems (EOLSS), because it provides a useful body of knowledge which should reach all peoples in their languages and in a way that fits their cultural backgrounds. UNL can do both: reproduce EOLSS knowledge in peoples' native languages, and enable them to explore it according to their cultural backgrounds. The UNL task is to make the entire EOLSS available in multiple languages starting with the six official languages of UNESCO. This task involves a two-step process: the first step is enconverting (encoding) the content of EOLSS from English into UNL (UNLization process); and the second is deconverting (decoding) EOLSS content from UNL into natural languages.

Combining Syntax, EBMT and a Transfer Approach to MT

Vandeghinste and Martens (2010) present another interesting option, in a number of papers in which the authors describe a system PaCo-MT that uses a transfer approach where syntax and examples are combined in a stochastic model. We quote from the Vandeghinste and Martens (2010) paper describing, "… the transfer component of a syntax-based Example-based Machine Translation system. The source sentence parse tree is matched in a bottom-up fashion with the source language side of a parallel example treebank, which results in a target forest … sent to the target language generation component." The novelty of the approach described in this paper was the bottom-up policy as opposed to the top-down one, in the choice of source sentence parse tree. Translations are example-based, in that "… as it uses a large set of translation examples (a parallel corpus) as training data to base its decisions on and it is *syntax-based* as the data in the parallel corpus is annotated with syntactic parse trees, both on the source and the target side. Input sentences are syntactically analysed, and the system generates target language parse trees where all ordering information is removed." The system uses a parser for the source language to parse the source side of the parallel corpus as well as the input sentence to feed the translation engine. The target language parser is only used to preprocess the target parallel corpus. The parallel treebank has also been commented upon in Tiedemann and Kotzé (2009), it is word aligned using GIZA++, and node aligned using a discriminative approach to tree alignment (Tiedemann 2011).

As the authors comment, using a syntax-based translation unit is like using a rule-based approach. In fact, the PaCo-MT system combines a stochastic example-based transfer system with the data-driven tree-to-tree based approach, transducing the source parse tree into a set of target language parse trees. This is done without node ordering, and reordering is done by a discriminative model for tree alignment. In this way, rule-based strengths are combined with PBSMT systems: in particular, the target tree-based language model is generated using a probabilistic context-free grammar

based on large monolingual treebanks which rather than reordering words or phrases, it addresses parse trees. We address more of these problems in the next section.

Syntax Based Approaches: From Hierarchical to SBSMT

Accurate translation may ensue from SMT and EBMT but there is no way to control the performance of such systems to obtain a 100% accurate translation all the time. So improvements may only come from external knowledge made available to the system either at runtime, producing some preprocessing and new models, or at the end of the computation, producing some postprocessing. For sometime the introduction of syntactic information in the training process did not seem to produce any improvement in the performance of STM. However, a number of papers appearing lately show that this is not always the case. In particular in language pairs which require heavy reordering, and/or have totally different grammatical structures, syntactic information seems particularly useful. The need for reordering in some language pairs is paramount and cannot be limited to local phrases. Syntax may provide means for an accurate reordering step. Syntax may also check for appropriate insertion of function words and their wordforms – in case of amalgamated function words like articulated prepositions in German and Romance languages.

- Language may have the problem of pro-dropping subject and object (like Japanese) or just subject as most Romance languages do;
- The most typical problem is semantic and word sense ambiguity that requires disambiguation: this may be done only by restricting the language model to a specific translation domain where the appropriate sense is usually easily capture. Or else a full-fledged words-sense disambiguation algorithm must be in place;
- Languages may use tenses differently or have more/less tenses – like perfect in English, and "imperfetto" in some Romance languages, simple past and "passato prossimo" versus "passato remoto" in Italian;
- Idioms may be difficult to trace in complete phrases (see Wehrli 2007, Wehrli et al. 2009) on the subject);
- Many transformations can be best explained in syntactic terms – see examples below;
- Syntactic annotation on the source input adds additional knowledge
- Syntactic annotation on the target output aids grammatical output

Here are some attempts at using syntax-based models:

- String to tree based translation systems (Yamada and Knight 2001; Galley et al. 2006; Marcu et al. 2006; Shen et al. 2008; Chiang et al. 2009)
- Using syntactic chunks (Schafer and Yarowsky 2003)
- Using syntactic features (Koehn and Knight 2003; Och and Ney 2003)

– Tree-to-string based translation systems (Quirk et al. 2005; Liu et al. 2006; Huang et al. 2006; Mi et al. 2008)
– Tree-to-tree based translation systems (Eisner 2003; Ding and Palmer 2005; Cowan et al. 2006; Zhang et al. 2007; Liu et al. 2009)

Early SMT syntactic models had worse results than PBSMT because phrase pairs limited to corresponding complete syntactic units were harmful for translation. Some of the advantages of SBMT are:

– Better overall handling of word order
– Better at translating discontinuous phrases (E.g. as X as Y- aussi X que Y)
– Especially advantageous for handling typologically different languages
– Fast and steady improvement in recent years

Syntax-based approaches for Machine Translation (MT) have gained popularity in recent times because of their ability to handle long distance reorderings (Wu 1997; Yamada and Knight 2002; Quirk et al. 2005; Chiang 2005), especially for divergent language pairs such as English-Hindi (or English-Urdu). Languages such as Hindi are also known for their rich morphology and long distance agreement of features of syntactically related units. Employing techniques that factor the lexical items into morphological factors can handle the morphological richness. The same applies to Arabic (see El Kholy and Habash 2010).

The first problem that SBMT aimed to solve was the issue of reordering, i.e. learning how to transform the sentence structure of one language into the sentence structure of another, in a way that is not tied to a specific domain or sub-domains, or indeed, sequences of individual words. An early attempt at greater generality in a purely phrasal setting was the alignment template approach (Och and Ney 2004). Newer approaches include formally syntactic (Chiang 2005), and linguistically syntactic approaches (Quirk et al. 2005; Huang et al. 2006; Wang et al. 2010).

The other fundamental issue SBMT targets, is extraposition and long distance movement which still pose a serious challenge to syntax-based machine translation systems. Even if the search algorithms could accommodate such syntactic discontinuities, we need appropriate models to account for such phenomena. Also if the system extracts extraposition templates, they may prove too sparse and brittle to accomodate the range of phenomena.

String models are popular in statistical machine translation. Approaches include word substitution systems (Brown et al. 1993), phrase substitution systems (Koehn et al. 2003; Och and Ney 2004), and synchronous context-free grammar systems (Wu and Wong 1998; Chiang 2005; Wong et al. 2005; Huang et al. 2009), all of which train on string pairs and seek to establish connections between source and target strings. By contrast, explicit syntax approaches seek to model directly the relations learned from parsed data, including models between source trees and target trees (Gildea 2003; Eisner 2003; Melamed 2004; Cowan et al. 2006), source trees and target strings (Quirk et al. 2005; Huang et al. 2006), or source strings and target trees (Yamada and Knight 2001; Galley et al. 2004). A strength of phrase models is that they can acquire all phrase pairs consistent with computed word alignments (Lopez

and Resnik 2006), concatenate those phrases together, and re-order them under several cost models. An advantage of syntax-based models is that outputs tend to be syntactically well-formed, with re-ordering influenced by syntactic context and function words introduced to serve specific syntactic purposes.

Generally speaking, syntactic models outperform string models for the simple reason that their output is still syntactically acceptable even for bad translations that may be semantically wrong, whereas the former produces bad translations that are also grammatically totally incorrect.

Hierarchical MT

In 2005 the first SMT system that uses hierarchical phrase-based decoding (HPBSMT) is presented (Chiang 2005), and is shown to improve the performance of phrase-based systems at least for all those concerned with Chinese. HPBSMT extends the PBSMT by allowing the use of non-contiguous phrase pairs. It incorporates reordering rules and in some way also the recursive structure of the sentence, implicitly adopting in this way a linguistic approach without including any linguistic representation of the data. To make the model sensitive to the syntax structure, a constituent feature was integrated into the translation model with the soft constraint method. It was defined as follows: it gains 1 for rules whose source side respect syntactic phrase boundary in the parse tree, and 0 otherwise. However, it did not achieve statistically significant improvement in the experiment. Marton and Resnik (2008) (hence M&R 2008) thought that different syntactic types may play different roles in the translation model. However, (Chiang 2005)'s method did not treat them discriminatively. They then defined soft constraint features for each constituent type based on the observation of this phenomenon. Their experiments showed that some constituent features significantly improved the performance, but others didn't. It is an interesting question whether all these constituent type models can work together efficiently. Although M&R 2008 did not give the experiments to support the positive answer. Chiang (2005) had already provided the evidence that their constituent models could not work together. (Chiang et al. 2008) thought one of its reasons were the limitations of MERT (Och 2003) with many features. We explore the topic of soft constraints more below.

HPBSMT is usually described as being formally similar to a syntactic model without linguistic commitments, in contrast with syntactic decoding which uses rules with linguistically motivated labels. However, as remarked in Hoang and Koehn (2010) – hence HK2010, the decoding mechanism for both hierarchical and syntactic systems are identical and the rule extraction are similar. Hierarchical and syntax statistical machine translation have made great progress in the last few years and now represent the state of the art in the field. Both use synchronous context free grammar (SCFG) formalism, consisting of rewrite rules which simultaneously parse the input sentence and generate the output sentence. The most common algorithm for decoding with SCFG is currently CKY+ with cube pruning, which works for

both hierarchical and syntactic systems, as implemented in Hiero (Chiang 2005), Joshua (Li et al. 2009), and Moses (Hoang et al. 2009). Again as commented by HK2010, simple HPBSMT have the advantage of ensuring broad coverage to their representations, but run the risk of using a rule for an inappropriate situation.

Most existing alignment methods simply consider a sentence as a sequence of words (Brown et al. 1993), and generate phrase correspondences using heuristic rules (Koehn et al. 2003). Some studies incorporate structural information into the alignment process *after* this simple word alignment (Quirk et al. 2005; Cowan et al. 2006). However, this is not sufficient because the basic word alignment itself is not good.

On the other hand, syntactic models have been proposed which use structural information from the beginning of the alignment process. Watanabe et al. (2000) and Menezes and Richardson (2001) proposed a structural alignment method. These methods use heuristic rules when resolving correspondence ambiguities. Yamada and Knight (2001) and Gildea (2003) proposed a tree-based probabilistic alignment methods. These methods reorder, insert or delete sub-trees on one side to reproduce the other side, but the constraints of using syntactic information is often too rigid. Yamada and Knight flattened the trees by collapsing nodes. Gildea cloned sub-trees to deal with the problem.

Rewrite rules in hierarchical systems have general applicability as their non-terminals are undecorated, giving hierarchical system broad coverage. However, rules may be used in inappropriate situations without the labeled constraints. The general applicability of undecorated rules create spurious ambiguity which decreases translation performance by causing the decoder to spend more time sifting through duplicate hypotheses. Syntactic systems make use of linguistically motivated information to bias the search space at the expense of limiting model coverage. The main problem to solve when using syntactic representation is the poor coverage of syntactically encoded translation rules and as a result the decoding phase has a low number of translation pairs.

Eventually, the ability to incorporate both source and target syntactic information in tree-to-tree models are believed to have a lot of potential to achieve promising translation quality. However, they are affected by rigid syntactic constraints and this may be the reason that conventional tree-to-tree based translation systems haven't shown superiority in empirical evaluations. We address more on this topic below.

Syntactic labels from parse trees can be used to annotate non-terminals in the translation model. This reduces incorrect rule application by restricting rule extraction and application. However, as noted in (Ambati and Lavie 2008) and elsewhere, the naive approach of constraining every non-terminal to a syntactic constituent severely limits the coverage of the resulting grammar. Therefore, several approaches have been used to improve coverage when using syntactic information. Zollmann and Venugopal 2006 allow rules to be extracted where non-terminals do not exactly span a target constituent. The non-terminals are then labeled with complex labels which amalgamates multiple labels in the span. This increases coverage at the expense of increasing data sparsity as the non-terminal symbol set increases dramatically.

Syntax-Based and Hierarchical Statistical MT

There are a great number of ways in which these two basic methods can be combined together and they will be reviewed below. Basically, what syntactic models do is explicitly to take into account the syntax of the sentences being translated. One simple approach is to limit the phrases learned by a standard PBSMT translation model to only those contiguous sequences of words that additionally correspond to constituents in a syntactic parse tree. However, a total reliance on such syntax-based phrases has been shown to be detrimental to translation quality, as the source-side and target-side tree structures heavily constrain the space of phrase segmentation of a parallel sentence. Noting that the number of phrase pairs extracted from a corpus is reduced by around 80% when they are required to correspond to syntactic constituents, Koehn et al. (2003) observed that many non-constituent phrase pairs that would not be included in a syntax-only model are in fact extremely important to system performance. Since then, researchers have explored effective ways for combining phrase pairs derived from syntax-aware methods with those extracted from more traditional PBSMT (see Xiong et al. 2010a). Briefly stated, the goal is to retain the high level of coverage provided by non-syntactic PBSMT phrases while simultaneously incorporating and exploiting specific syntactic knowledge.

At the same time, it is desirable to include as much syntactic information in the system as possible in order to carry out linguistically motivated reordering, for example: an extended and modified version of the approach of Tinsley et al. (2007), i.e. extracting syntax-based phrase pairs from a large parallel parsed corpus, combining them with PBSMT phrases, and performing joint decoding in a syntax-based MT framework without loss of translation quality. This effectively addresses the low coverage of purely syntactic MT without discarding syntactic information.

A lot of work has focused on combining hierarchical and syntax translation, utilizing the high coverage of hierarchical decoding and the insights that syntactic information can bring. This is done with the aim to balance the generality of using undecorated non-terminals with the specificity of labeled non-terminals. In particular, systems can use syntactic labels from a source language parser to label non-terminal in production rules. However, other source span information, such as chunk tags, can also be used, as will be discussed below.

Researchers have experimented with different methods for combining the hierarchical and syntactic approaches. Syntactic translation rules are used concurrently with a hierarchical phrase rules by training them independently and then using them concurrently to decode sentences.

Another possible method is to use one translation model containing both hierarchical and syntactic rules. Moreover, rules can contain both decorated syntactic non-terminals, and undecorated hierarchical-style non-terminals (in addition, the left-hand-side non-terminal may, or may not be decorated). Improvements may come by using simpler tools: for instance linguistic information coming from shallow parsing techniques – like the chunk tagger (Abney 1991) instead of a full-fledged parser-rule extraction to reduce spurious ambiguity.

Zollmann and Venugopal 2006 etc. overcome the restrictiveness of the syntax-only model by starting with a complete set of phrases as produced by traditional PBSMT heuristics, then annotating the target side of each phrasal entry with the label of the constituent node in the target-side parse tree that subsumes the span. They then introduce new constituent labels to handle the cases where the phrasal entries do not exactly correspond to the syntactic constituents. Liu et al. (2006) also add non-syntactic PBSMT phrases into their tree-to-string translation system.

There has been much effort to improve performance for hierarchical phrase-based machine translation by employing linguistic knowledge. For instance M&R 2008 etc., explore "soft syntactic constraints" on hierarchical phrase model; (Stein et al. 2010) focus on syntactic constraints not only via the constituent parse but also via the dependency parse tree of source or target sentence. (Chiang et al. 2009; Chiang 2010) similarly define many syntactic features including both source and target sides but integrate them into the translation model by MIRA algorithm to optimize their weights.

In particular, M&R 2008 extend a hierarchical PBSMT system with a number of features to prefer or disprefer certain types of syntactic phrases in different contexts. Restructuring the parse trees to ease their restrictiveness is another recent approach: in particular, Chao Wang et al. (2007) binarize source-side parse trees in order to provide phrase pair coverage for phrases that are partially syntactic. Tinsley et al. (2007) showed an improvement over a PBSMT baseline on four tasks in bidirectional German–English and Spanish–English translation by incorporating syntactic phrases derived from parallel trees into the PBSMT translation model. They first word align and extract phrases from a parallel corpus using the open-source Moses PBSMT toolkit (Koehn et al. 2007), which provides a baseline SMT system. Then, both sides of the parallel corpus are parsed with independent automatic parsers, subtrees from the resulting parallel treebank are aligned, and an additional set of phrases (with each phrase corresponding to a syntactic constituent in the parse tree) is extracted. The authors report statistically significant improvements in translation quality, as measured by a variety of automatic metrics, when the two types of phrases are combined in the Moses decoder.

ISI's system obtained best performance on *Ch_En* at NIST 2009. However there are also drawbacks in using this approach and they are all related to the difficulty inherent in producing the needed representation which require language-specific resources (parsers, morphological analysers, etc.). Since parsing is by itself also far from reaching 100% accuracy, the performance of the SBMT system is heavily dependent on parsing quality. Also due to the need to encode additional information to the one represented by simple words, the system will need larger search space and will result in overall costlier processing. Other limitations with the syntax based approaches (such as Quirk et al. 2005; Chiang 2005) are, that they do not offer flexibility for adding linguistically motivated features, and that it is not possible to use morphological factors in the syntax based approaches. In general, the translation quality has shown improvements: in particular, these improvements are due to the more accurate phrase boundary detection. So we may safely say that syntactic phrases are a much more precise representation of translational equivalence, and this is the main reason for adopting such an approach.

Introducing Soft Syntactic Features with Discriminative Classifiers

In the last decade, there has been countless research in soft syntactic features, much of which has led to the improved performance for Hiero. However, it seems that all the syntactic constituent features cannot efficiently work together in the Hiero optimized by MERT. So a more general soft syntactic constraint model has been proposed, based on discriminative classifiers for each constituent type and integrate all of them into the translation model with a unified form. The experimental results show that this method significantly improves the performance on the NIST05 Chinese-to-English translation task.

Soft Syntactic Constraint models (SSC) have been proposed at first by M&R 2008 as heuristic models, while SSC models proposed by Liu et al. (2011) are much more general and based on discriminative classifiers. In this latter paper, they further decompose crossing constituents into three types to contain more syntactic information. For example, similarly to Zollmann and Venugopal 2006, the crossing constituent "NP+" is divided into L\NP, NP/R, and L\NP/R, which means a partial syntactic category NP missing some category to the left, the right and the left and right together, respectively. They are called *general constituent labels (GCL)*. Chiang et al. 2008 introduce heuristic models, that are not sensitive to other features such as boundary word information. However, (Xiong et al. 2006), showed in previous work that these features are helpful for the translation model. On the other hand, uniform combination of all the constituent models may cause a model bias, since some constituent types occur more often than others.

Liu et al. (2011) propose a discriminative soft constraint model for each syntactic constituent type. The underlying idea is to improve the model by integrating it with context information. They consider several classifiers with different accuracy to construct soft constraint models, and they aim to study the effect of the accuracy of the classifiers on the translation performance. Then, they investigate an efficient method to combine all the models to give a unified soft constraint model. Instead of uniformly combining all the models, they introduce a prior distribution for them and combine them with the priority.

The authors propose a unified SSC model based on discriminative classifiers for hierarchical phrase-based translation. Experimental results prove the effectiveness of the method on the NIST05 Chinese-to-English translation task. The experiment shows that the discriminative soft syntactic constraint model achieves better result over the heuristic model of M&R 2008; then, it empirically proves that the more accurate classifier can gain better results when building a sub-model for the translation model. Finally we have an efficient method which integrates all models with respect to general constituent labels into hierarchical phrase translation model and improves its performance.

For different syntactic categories (e.g. NP), M&R 2008 defined some kinds of soft-constraint constituency features (e.g. NP=, NP+, NP_, etc.) for Hiero rules. For instance, if a synchronous rule is used in a derivation, and the span of is a cross constituent "NP+" in the source language parse tree, this rule will get an additional value to the model

score for the case of "NP+". In fact, each of these features can also be viewed as a discrete model with value {0, 1}, i.e. for the case of "NP=" if the span of is exactly "NP", the rule gets a score 1 and 0 otherwise. These constituency features don't distinguish the rules with the same span in the source language. For a training instance corresponding to a rule, inspired by previous work (Zollmann and Venugopal 2006; He et al. 2008; Cui et al. 2010), they design the following features to train SSC models;

Syntactic features, which are the general constituent labels defined in section "Specific Issues in Hybrid MT" for the spans of r and the nonterminal symbols in the source side.
Parts-of-speech (POS) features, which are the POS of the words immediately to the left and right of and those of the boundary words covered by the nonterminal symbols in the source side.
Length features, which are the length of sub-phrases covered by the nonterminal symbols in the source side.

In fact, the models can be extended to include other features, especially those in the target side. In order to compare these models with the work of M&R 2008, they merely introduce several features. They implement a hierarchical phrase-based system as the baseline, similar to Hiero (Chiang 2005), and use XP (M&R 2008) as the comparison system. They use the default setting as Hiero. Word alignment for each sentence pair is obtained as usual. Then, Stanford parser (Klein and Manning 2003) is employed to generate the parse tree for the source side of the data. They acquire about 15.85M training examples among which are 6.81M positive and 9.04M negative examples respectively. There are 88 general constituent labels in all. They employ the open toolkits of MaxEnt and LogReg to train SSC models for each GCL, and construct a linear combination model with them, where the interpolation weight is set to 0.86. They train a 4-gram language model on the Xinhua portion of the English Gigaword corpus using the SRILM Toolkits (Stolcke 2002) with modified Kneser-Ney smoothing (Chen and Goodman 1998). In the experiments, case-sensitive BLEU4 metric (Papineni et al. 2002) measures the translation performances and the statistical significance in BLEU score differences is tested by paired bootstrap re-sampling (Koehn 2004).

Translation Consistency Enforced by Graph-Based Learning

Alexandrescu and Kirchhoff (2009) propose a new graph-based learning algorithm is proposed with structured inputs and outputs to improve consistency in phrase-based statistical machine translation. They define a joint similarity graph over training and test data and use an iterative label propagation procedure to regress a scoring function over the graph. For the purpose of reranking, the resulting scores for unlabeled samples (translation hypotheses) are then combined with standard model scores in a log-linear translation model. From a machine learning perspective, graph-based learning (GBL) is applied to a task with structured inputs and outputs. This is a novel contribution

in itself since previous applications of GBL have focused on predicting categorical labels. The evaluation demonstrates significant improvements over the baseline.

As discussed above, current phrase-based statistical machine translation (SMT) systems commonly operate at the sentence level; each sentence is translated in isolation, even when the test data consists of internally coherent paragraphs or stories, such as news articles. For each sentence, SMT systems choose the translation hypothesis that maximizes a combined log-linear model score, which is computed independently of all other sentences, using globally optimized combination weights. Thus, similar input strings may be translated in very different ways, depending on which component model happens to dominate the combined score for that sentence. A phrase can be translated differently – and wrongly – due to different segmentations and phrase translations chosen by the decoder. Though different choices may be sometimes appropriate, the lack of constraints enforcing translation consistency often leads to suboptimal translation performance. It would be desirable to counter this effect by encouraging similar outputs for similar inputs (under a suitably defined notion of similarity, which may include, for example, a context specification for the phrase/sentence). In machine learning, the idea of forcing the outputs of a statistical learner to vary smoothly with the underlying structure of the inputs has been formalized in the graph-based learning (GBL) framework. In GBL, both labeled (train) and unlabeled (test) data samples are jointly represented as vertices in a graph whose edges encode pairwise similarities between samples. Various learning algorithms can be applied to assign labels to the test samples while ensuring that the classification output varies smoothly along the manifold structure defined by the graph. GBL has been successfully applied to a range of problems in computer vision, computational biology, and natural language processing. However, in most cases, the learning tasks consisted of unstructured classification, where the input was represented by fixed length feature vectors and the output was one of a finite set of discrete labels. In machine translation, by contrast, both inputs and outputs consist of word strings of variable length, and the number of possible outputs is not fixed and practically unlimited.

GBL is an instance of semi-supervised learning, specifically transductive learning. A different form of semi-supervised learning (self-training) has been applied to MT by (Ueffing et al. 2007; Fraser and Marcu 2006). This is the first study to explore a graph-based learning approach. In the machine learning community, work on applying GBL to structured outputs is beginning to emerge. The graph-based learning scheme is used to implement a consistency model for SMT that encourages similar inputs to receive similar outputs. Evaluation on two small-scale translation tasks showed significant improvements of up to 2.6 points in BLEU and 2.8% PER. As the authors report, the approach needs improvements in future work that will include testing different graph construction schemes, in particular better parameter optimization approaches and better string similarity measures; always according to the authors, more gains can be expected when using better domain knowledge in constructing the string kernels. This may include e.g. similarity measures that accommodate POS tags or morphological features, or comparisons of the syntax trees of parsed sentence. The latter could be quite easily incorporated into a string kernel or the related tree kernel similarity measure.

Problems in Combining PBSTM and SBSTM: Rules and Constraints

Galley et al. (2004) create minimal translation rules which can explain a parallel sentence pair but the rules generated are not optimized to produce good translations or coverage in any SMT system. This work was extended and described in (Galley et al. 2006) who create rules composed of smaller, minimal rules, as well as deal with unaligned words. These measures are essential for creating good SMT systems, but again, a parser strictly constrains the rules of syntax.

DeNeefe (2007: 756, 757) proposed the GHKM Galley's – where GHKM is an acronym for the authors names Galley, Hopkins, Knight and Marcu – syntax-based extraction method for learning statistical syntax-based translation rules, presented first in (Galley et al. 2004) and expanded on in (Galley et al. 2006). It is similar to phrase-based extraction in that it extracts rules consistent with given word alignments. A primary difference is the use of syntax trees on the target side, rather than sequences of words. The basic unit of translation is the translation rule, consisting of a sequence of words and variables in the source language, a syntax tree in the target language having words or variables at the leaves, and again a vector of feature values which describe this pair's likelihood. Translation rules can:

- Look like phrase pairs with syntax decoration
- Carry extra contextual constraints
- Be non-constituent phrases
- Contain non-contiguous phrases, effectively "phrases with holes"
- Be purely structural (no words)
- Re-order their children

Decoding with this model produces a tree in the target language, bottom-up, by parsing the foreign string using a CYK parser (Chappelier and Rajman 1998) and a binarized rule set (Zhang et al. 2008). During decoding, features from each translation rule are combined with a language model using a log-linear model to compute the score of the entire translation. The GHKM extractor learns translation rules from an aligned parallel corpus where the target side has been parsed. This corpus is conceptually a list of tuples of 'source sentence, target tree, bi-directional word alignments' which serve as training examples. For each training example, the GHKM extractor computes the set of minimally-sized translation rules that can explain the training example while remaining consistent with the alignments. This is, in effect, a non-overlapping tiling of translation rules over the tree-string pair. If there are no unaligned words in the source sentence, this is a unique set. This set, ordered into a tree of rule applications, is called the derivation tree of the training example. As with ATS (Alignment Template System), translation rules are extracted and counted over the entire training corpus, a count of one for each time they appear in a training example. These counts are used to estimate several features, including maximum likelihood probability features.

To extract all valid tree-to-tree rules, (Liu et al. 2009) extends the famous tree-to-string rule extraction algorithm GHKM (Galley et al. 2004) to their forest-based

tree-to-tree model. However, only with GHKM rules, the rule coverage is very low. As SPMT rules (Marcu et al. 2006) have proven to be a good complement to GHKM (DeNeefe et al. 2007), Zhai et al. also extract full lexicalized SPMT (Marcu et al. 2006: the acronym stands for"Statistical machine translation with syntactified target language phrases") rules to improve the rule coverage.

The tree-to-tree style SPMT algorithm used in their experiments is described as follows:

> … for each phrase pair, traverse the source and target parsing tree bottom up until it finds a node that subsumes the corresponding phrase respectively, then extract a rule whose roots are the nodes just found and the leaf nodes are the phrases.

However, even with GHKM and SPMT rules, the rule coverage is still very low since tree-to-tree models require that both source side and target side of its rule must be a subtree of the parsing tree. With this hard constraint (Liu et al. 2009; Chiang 2010), the model would lose a large amount of bilingual phrases which are very useful to the translation process (DeNeefe et al. 2007). In particular it can be shown that phrase-based models can extract all useful phrase pairs, while string-to-tree and tree-to-string model can only extract part of them because of the one-side subtree constraint. Further, with the rigid ***both-side subtree constraint*****,** the rule space of tree-to-tree model is the narrowest, accounting only for at most 8.45% of all phrase pairs. Hence, learning to enlarge the rule coverage is the challenge for tree-to-tree models.

In the decoding process, the procedure traverses the source parsing tree in a bottom up fashion and tries to translate the subtree rooted at the current node. If the employed rule is full lexicalized, *candidate translations* are generated directly. Otherwise new candidate translations are created by combining target terminals of the rule and candidate translations of the corresponding descendant nodes of the current node. Root node of the parsing tree will be the last visited node and the best translation is chosen as usual from its best candidate translations. Broadly, tree-to-tree based decoding is node-based, i.e., only the source spans governed by tree nodes can be translated as a unit. These spans are called ***translation spans***. During decoding, translation spans are used for translation, while other spans are ignored completely even if they include better translations. Thus this rigid constraint (they call it ***node constraint)*** will exclude many good translations. Zhai et al. (2011) use the Chinese part of the FBIS corpus as a test set: in their statistics, there are in total of 14.68M effective translation spans in the corpus. However, only 44.6% (6.54M spans) of them are governed by tree nodes. This low proportion would definitely lead to an exceptionally narrow search space for tree-to-tree model and a poor translation quality.

In addition, the model is also heavily affected by the ***exact matching constraint*** which means only the rules completely matching part of the source tree structure can be used for decoding. Since parsing errors are very common with automatic parsers, the mismatch is not rare. Moreover, the large and flat structures which have a close relation with reordering are also hard to match exactly. Thus with such constraint, many rules cannot be employed during decoding even if by the model extracts them and the search space is necessarily decreased.

In order to resolve the constraints, two simple but very effective approaches are proposed: (1) integrating bilingual phrases to improve the rule coverage problem; (2) binarizing the bilingual parsing trees to relieve the rigid syntactic constraints. Other systems using transducers with MLE probabilities may also benefit from additional reordering models (more on this topic below).

Huang et al. (2010) decorate the syntax structure into the non-terminal in hierarchical rules as a feature vector. During decoding time, they calculate the similarity between the syntax of the source side and the rules used to derive translations, and then they add the similarity measure to translation model as an additional feature. They don't directly use the syntax knowledge to calculate the additional feature score, but use it to derive a latent syntactic distribution. He et al. (2008) and Cui et al. (2010) employ the syntax knowledge as some of the features to construct rule selection models. When training discriminative models training examples are derived from the rule extraction or from the formal bilingual parsing derivation forest of the training data. Their strong results reinforce the claim that discriminative models are useful in building the sub-model in translation.

Huang and Chiang (2007) use parse information of the source language, and their production rules consist of source tree fragments and target languages strings. During decoding, a packed forest of the source sentence is used as input, and the production rule tree fragments are applied to the packed forest. Liu et al. (2009) use joint decoding with a hierarchical and tree-to-string model and find that translation performance increases for a Chinese-English task.

Others have sought to add soft linguistic constraints to hierarchical models using addition feature functions, such as M&R 2008 who add feature functions to penalize or reward non-terminals which cross constituent boundaries of the source sentence. Shen et al. (2009) discuss soft syntax constraints and context features in a dependency tree translation model. The POS tag of the target head word is used as a soft constraint when applying rules. Also, a source context language model and a dependency language model are used as features. Most SMT systems use the Viterbi approximation whereby the derivations in the log-linear model are not marginalized, but the maximum derivation is returned. String-to-tree models build on this so that the most probable derivation, including syntactic labels, is assumed to be the most probable translation. This fragments the derivation probability and further partitions the search space, leading to pruning errors. Venugopal et al. (2009) attempts to address this by efficiently estimating the score over an equivalent unlabeled derivation from a target syntax model. Ambati and Lavie (2008) and Ambati et al. (2009) note that tree-to-tree often underperforms models with parse tree only on one side due to the non-isomorphic structure of languages. This motivates the creation of an isomorphic backbone into the target parse tree, while leaving the source parse unchanged.

Hoang and Koehn (2010), present a new translation model that includes undecorated hierarchical-style phrase rules, decorated source-syntax rules, and partially decorated rules. Results show an increase in translation performance of up to 0.8% BLEU for German–English translation when trained on the news-commentary corpus, using syntactic annotation from a source language parser. Also experimenting with annotation from shallow taggers may increase BLEU scores.

This continues earlier work in (Chiang 2005) but they see gains when finer grain feature functions are used. The weights for feature function is tuned in batches due to the deficiency of MERT when presented with many features. Chiang et al. (2008) rectified this deficiency by using the MIRA to tune all feature function weights in combination. However, the translation model continues to be hierarchical. Chiang et al. (2009) added thousands of linguistically-motivated features to hierarchical and syntax systems, However, the source syntax features are derived from the research above. The translation model remains constant but the parameterization changes.

Syntax Based SMT and Fuzzy Methods

Tree-to-tree translation models suffer from unsatisfactory performance due to the limitations both in rule extraction and decoding procedure, and in several rigid syntactic constraints that severely hamper these models. These constraints include: the both-side subtree constraint in rule extraction, the node constraint and the exact matching constraint in decoding. Zhai et al. (2011) propose two simple but effective approaches to overcome the constraints: utilizing fuzzy matching and category translating to integrate bilingual phrases and using head-out binarization to binarize the bilingual parsing trees. Their experiments show that the proposed approaches can significantly improve the performance of tree-to-tree system and outperform the state-of-the-art phrase-based system Moses.

Two main directions have emerged to overcome the limitations discussed above. One is to loose the syntactic constraints. (Zhang et al. 2008) proposes a *tree-sequence based tree-to-tree model* that represents rules with tree sequences and takes all spans as translation spans. This method resolves the both-side subtree constraint and the node constraint thoroughly, but it neglects the bad influence of the exact matching constraint. Furthermore, it is obvious that each bilingual phrase would multiply into many tree sequence rules with different structures, which definitely leads to serious rule expansion to increase the decoding burden. In the other direction, more information is introduced into the model. (Liu et al. 2009) substitutes one-best tree with packed forest for tree-to-tree model which can compactly encode many parses and successfully relieve the constraints. But even with packed forest, the rule coverage is still very low. The two directions have proven to outperform their conventional counterparts significantly. However, whether tree sequence or packed forest, they are all complicated to deal with in decoding stage, and furthermore, they both need to modify the conventional tree-to-tree model. Thus they must heavily adjust the original decoding algorithm to cater for the corresponding changes.

To improve the conventional tree-to-tree model the authors propose integrating bilingual phrases and binarizing the bilingual parsing trees. (Liu et al. 2006) and (Mi et al. 2008) utilize bilingual phrases to improve tree-to-string and forest-to-string model. Other authors integrate bilingual phrases into tree-to-tree model to resolve the problem of poor coverage of rules. Of the two, this model is the more

difficult since it must provide syntactic structures for both the source and target phrases to serve the decoding process of the model.

In traditional tree-to-tree based decoding, source side of the rule is employed to match the source parsing tree exactly. Thus if we want to use a source phrase, theoretically we must decorate it with the corresponding syntax structure like the tree-sequence based model. However, it has been shown that exact match would do harm to the translation quality. Thus instead of syntax structures, source phrases are decorated with syntactic categories which are necessary and effective for translation (Zhang et al. 2011). When decoding with these source phrases, the system ignores the internal structure of the subtree for translation and only matches the rule's category with root node of the subtree along with the matching between leaf nodes. Normally, if the system tries an exact match, a given rule may not be employed in case of mismatch between categories of rule and tree structure. Hence, to maximize the capacities of the source phrases, the fuzzy matching method can be employed which has been successfully employed in hierarchical phrase-based model (Huang et al. 2010) and string-to-tree model (Zhang et al. 2011) to match categories. With fuzzy matching method, Zhai et al. (2011) represent each SAMT-style syntactic category with a real-valued vector $F_{(c)}$ using latent syntactic distribution. That is to say, they transform an original source phrase by decorating it with a SAMT-style syntactic category and a corresponding real-valued vector. During decoding, they consider all possible source phrases and compute the similarity scores between categories of phrases and head nodes of the current translated structure. Then the similarity score will serve as a good feature *(similarity score feature)* incorporated into the model and will let it learn how to respect the source phrases.

Combining PBSTM and SBSTM but Then Syntax-Prioritizing

A key concern in building syntax-based machine translation systems is how to improve coverage by incorporating more traditional phrase-based SMT phrase pairs that do not correspond to syntactic constituents. Improved precision due to the inclusion of syntactic phrases can be seen by examining a translation example and the phrasal chunks chosen which exist in the baseline PBSMT phrase table, but do not make it into the top-best translation in the PBSMT-only scenario because of its high ambiguity factor. Hanneman and Lavie (2009) propose an approach which is structurally similar to that of Tinsley et al. (2007), extended or modified in a number of key ways. At first, they extract both non-syntactic PBSMT and syntax-driven phrases from a parallel corpus that is two orders of magnitude larger. Then, they apply a different algorithm for subtree alignment, proposed by Lavie et al. (2008), which proceeds bottom-up from existing statistical word alignments, rather than inducing them top-down from lexical alignment probabilities. In addition to combining straightforwardly syntax-derived phrases with traditional PBSMT phrases, they propose a new combination technique that removes PBSMT phrases whose source-language strings are already covered by a syntax-derived phrase. This new

syntax-prioritized technique results in a 61% reduction in the size of the combined phrase table with only a minimal decrease in automatic translation metric scores. Finally, and crucially, they carry out the joint decoding over both syntactic and non-syntactic phrase pairs in a syntax-aware MT system, which allows a syntactic grammar to be put in place on top of the phrase pairs to carry out linguistically motivated reordering, hierarchical decoding, and other operations.

A small number of grammar rules are then used to correct the structure of constituents which require some reordering in the sentence. After inspecting the output of the test set they find that the grammar is 97% accurate in its applications, making helpful reordering changes 88% of the time.

The statistical transfer ("Stat-XFER") framework (Lavie 2008; and recent extension by Ambati and Lavie 2008) is the base MT system used for an experiment that we report here. It is similar to what we already discussed under section "Combining Syntax, EBMT and a Transfer Approach to MT" making exception for the stochastic Example-Based approach. The core of the framework is a transfer engine using two language-pair-dependent resources: a grammar of weighted synchronous context-free rules, and a probabilistic bilingual lexicon. Once the resources have been provided, the Stat-XFER framework carries out translation in a two-stage process, first applying the lexicon and grammar to parse synchronously an input sentence, then running a monotonic decoder over the resulting lattice of scored translation pieces assembled during parsing to produce a final string output (see Dyer et al. 2008). Reordering is applied only in the first stage, driven by the syntactic grammar; the second-stage monotonic decoder only assembles translation fragments into complete hypotheses. Each Stat-XFER bilingual lexicon entry has a synchronous context-free grammar (SCFG) expression of the source- and target-language production rules. The SCFG backbone may include lexicalized items, as well as non-terminals and pre-terminals from the grammar. Constituent alignment information specifies one-to-one correspondences between source-language and target-language constituents on the right-hand side of the SCFG rule. Rule scores for grammar rules, if they are learned from data, are calculated in the same way as the scores for lexical entries. The grammar and lexicon are extracted from a large parallel corpus that has been statistically word-aligned and independently parsed on both sides with automatic parsers. Word-level entries for the bilingual lexicon are directly taken from word alignments; corresponding syntactic categories for the left-hand side of the SCFG rules are obtained from the preterminal nodes of the parse trees. Phrase-level entries for the lexicon are based on node-to-node alignments in the parallel parse trees. In the straightforward "tree-to-tree" scenario, a given node ns in one parse tree S will be aligned to a node nt in the other parse tree T if the words in the yield of ns are all either aligned to words within the yield of nt or have no alignment at all. If there are multiple nodes nt satisfying this constraint, the node in the tree closest to the leaves is selected. Each aligned node pair (ns,nt) produces a phrase-level entry in the lexicon, where the left-hand sides of the SCFG rule are the labels of ns and nt, and the right-hand sides are the yields of those two nodes in their respective trees. In the expanded "tree-to-tree-string" configuration, if no suitable node nt exists, a new node n′s is introduced into T as a projection of ns, spanning the yield of the words in T aligned to the yield of ns.

Conceptually, they take the opposite approach to that of Tinsley et al. (2007) by adding traditional PBSMT phrases into a syntax-based MT system rather than the other way around. They begin by running steps 3 through 5 of the Moses training script (Koehn et al. 2007), which results in a list of phrase pair instances for the same word-aligned corpus to which they applied the syntax-based extraction methods. Given the two sets of phrases, they explore two methods of combining them: direct combination and syntax-prioritized combination.

- **Direct Combination.** Following the method of Tinsley et al. (2007), they directly combine the counts of observed syntax-based phrase pairs with the counts of observed PBSMT phrase pairs. This results in a modified probability model in which a higher likelihood is moved onto syntactic phrase pairs that were also extractable using traditional PBSMT heuristics. It also allows either extraction mechanism to introduce new entries into the combined phrase table that were not extracted by the other, thus permitting the system to take full advantage of complementary information provided by PBSMT phrases that do not correspond to syntactic constituents.
- **Syntax-Prioritized Combination.** Under this method, they take advantage of the fact that syntax-based phrase pairs are likely to be more precise translational equivalences than traditional PBSMT phrase pairs, since constituent boundaries are taken into account during phrase extraction. PBSMT phrases whose source-side strings are already covered by an entry from the syntactic phrase table are removed; the remaining PBSMT phrases are combined as in the direct combination method above. The effect on the overall system is to trust the syntactic phrase pairs in the cases where they exist, supplementing with PBSMT phrase pairs for non-constituents.

Syntax MT and Dependency Structures

Hoang and Koehn (2010) present an experiment which shows how both hierarchical and syntax-based SMT can be used fruitfully to improve the performance of a system. Japanese and Chinese are the two languages mostly involved in experimenting with syntax-based MT, in particular due to structural differences between the two languages and English. Nakazawa and Kurohashi (2011) introduce a tree-based reordering model which models word or phrase dependency relations in dependency tree structures of source and target languages. They propose a phrase alignment method which models word or phrase dependency relations in dependency tree structures of source and target languages. For a pair of correspondences which has a parent–child relation on one side, the dependency relation on the other side is defined as the relation between the two correspondences. It is a kind of tree-based reordering model, and can capture non-local reorderings which sequential word-based models often cannot handle properly. The model is also capable of estimating phrase correspondences automatically without heuristic rules. The model is trained in two steps: Step 1 estimates word translation probabilities, and Step 2 estimates

phrase translation probabilities and dependency relation probabilities. Both Step 1 and Step 2 are performed iteratively by the EM algorithm. During the Step 2 iterations, word correspondences are grown into phrase correspondences.

Experimental results of alignment show that the model could achieve F-measure 1.7 points higher than the conventional word alignment model with symmetrization algorithms.

The authors consider that there are two important needs in aligning parallel sentences written in very different languages such as Japanese and English. One is to adopt structural or dependency analysis into the alignment process to overcome the difference in word order. The other is that the method needs to have the capability of generating phrase correspondences, that is, one-to-many or many-to-many word correspondences.

Nakazawa and Kurohashi (2008) also proposed a model focusing on the dependency relations. Their model has the constraint that content words can only correspond to content words on the other side, and the same applies for function words. This sometimes leads to an incorrect alignment. Thus they have removed this constraint to make more flexible alignments possible. Moreover, in their model, some function words are brought together, and thus they cannot handle the situation where each function word corresponds to a different part. The smallest unit of our model is a single word, which should solve this problem.

Chang et al. (2011) note that structural differences between Chinese and English are a major factor in the difficulty of machine translation from Chinese to English. The wide variety of such Chinese-English differences include the ordering of head nouns and relative clauses, and the ordering of prepositional phrases and the heads they modify. Previous studies have shown that using syntactic structures from the source side can help MT performance on these constructions. Most of the previous syntactic MT work has used phrase structure parses in various ways, either by doing syntax-directed translation to translate directly parse trees into strings in the target language (Huang et al. 2006), or by using source-side parses to preprocess the source sentences (Wang et al. 2007). One intuitive solution for using syntax is to capture different Chinese structures that might have the same meaning and hence the same translation in English. But it turns out that phrase structure (and linear order) are not sufficient to capture this meaning relation. Two sentences with the same meaning can have different phrase structures and linear orders. They propose to use *typed dependency* parses instead of phrase structure parses. Typed dependency parses give information about grammatical relations between words, instead of constituency information. They capture syntactic relations, such as *nsubj* (nominal subject) and *dobj* (direct object),, but also encode semantic information such as in the *loc* (localizer) relation. This suggests that this kind of semantic and syntactic representation could have more benefit than phrase structure parses. Chinese typed dependencies are automatically extracted from phrase structure parses. In English, this kind of typed dependencies has been introduced by de Marneffe and Manning (2008) and de Marneffe et al. (2006). Using typed dependencies, it is easier to read out relations between words, and thus the typed dependencies have been used in meaning extraction tasks. Features over the Chinese typed dependencies are used in a phrase-based MT system when deciding

whether one chunk of Chinese words (MT system statistical phrase) should appear before or after another. To achieve this, a discriminative phrase orientation classifier is trained following the work by Zens and Ney (2006), and the system uses grammatical relations between words as extra features to build the classifier. Then the phrase orientation classifier is applied as a feature in a phrase-based MT system to help reordering. Basic reordering models in phrase-based systems use linear distance as the cost for phrase movements (Koehn et al. 2003). The disadvantage of these models is their insensitivity to the content of the words or phrases. More recent work (Tillman 2004; Och & Ney 2004; Koehn et al. 2007) has introduced lexicalized reordering models which estimate reordering probabilities conditioned on the actual phrases. Lexicalized reordering models have brought significant gains over the baseline reordering models, but one concern is that data sparseness can make estimation less reliable. Zens and Ney (2006) proposed a discriminatively trained phrase orientation model and evaluated its performance as a classifier and when plugged into a phrase-based MT system. Their framework allows us easily to add in extra features. Therefore it is used as a testbed to see if features from Chinese typed dependency structures can effectively be used to help reordering in MT.

The target language (English) translation is built from left to right. The phrase orientation classifier predicts the start position of the next phrase in the source sentence. They use the simplest class definition and group the start positions into two classes: one class for a position to the left of the previous phrase (reversed) and one for a position to the right (ordered). The basic feature functions are similar to what Zens and Ney (2006) used in their MT experiments. The basic binary features are source words within a window of size 3 around the current source position *j,* and target words within a window of size 3 around the current target position *i.* Yhe classifier experiments in Zens and Ney (2006) also uses word classes to introduce generalization capabilities. In the MT setting it's harder to incorporate the part-of-speech information on the target language. Zens and Ney (2006) also exclude word class information in the MT experiments. In the work they also use word features as basic features for the classification experiments. Assuming the Chinese sentence to translate has been parsed and grammatical relations in the sentence have been extracted, the path between the two words annotated by the grammatical relations is used. This feature helps the model learn the relation between the two chunks of Chinese words. The feature is defined as follows: for two words at positions p and q in the Chinese sentence ($p<q$), find the shortest path in the typed dependency parse from p to q, concatenate all the relations on the path and use that as a feature.

Cherry and Lin (2003) proposed a model which uses a source side dependency tree structure and constructs a discriminative model. However, there is the defect that its alignment unit is a word, so it can only find one-to-one alignments. On the contrary, when aligning very different language pairs, the most important need is the capability of generating both one-to-many and many-to-many correspondences.

Venkatapathy et al. (2010) propose an English-Hindi dependency-based statistical system that uses discriminative techniques to train its parameters. The use of syntax (dependency tree) allowed them to address the large word-reorderings between English and Hindi. And, discriminative training allows us to use rich feature sets, including linguistic features that are useful in the machine translation task.

Morphological decomposition is useful where there is very limited parallel corpora available, and breaking words into smaller units helps in reducing sparsity. In order to handle phenomena, such as long-distance word agreement to achieve accurate generation of target language words, the inter-dependence between the factors of syntactically related words needs to be modeled effectively.

Knowledge-Based MT Systems

Our focus in this section will be knowledge-based systems, i.e. systems which are a combination of both syntax and semantic knowledge to inform statistical models and learning. In particular, I assume that both syntax and semantics should also inform automatic evaluation in order to improve precision. Semantics in this case refers to ontologies like SUMO or WORDNET, but then, in order to be effective, should also include some Word-Sense Disambiguation or at least semantic similarity processing step. Other recent procedures for assessing – and evaluating – semantic similarity are based on Text Entailment techniques, but are less frequently used. Taxonomies and ontologies are data structures that organise conceptual information by establishing relations among concepts, hierarchical and partitive relations being the most important ones. One of the first idea was that of using a multilingual ontology as an interlingua (Hovy and Nirenburg 1992; Hovy 1998; Hovy et al. 2006; Philpot et al. 2010). Nowadays, ontologies have a wide range of uses in many domains, for example, finance (International Accounting Standards Board 2007), bio-medicine (Collier et al. 2008; Ashburner et al. 2000) and libraries (Mischo 1982). These resources normally attach labels in natural language to the concepts and relations that define their structure, and these labels can be used for a number of purposes, such as providing user interface localization (McCrae et al. 2011), multilingual data access (Declerck et al. 2010), information extraction (Müller et al. 2004) and natural language generation (Bontcheva 2005). Applications that use such ontologies and taxonomies will require translation of the natural language descriptions associated with them in order to adapt these methods to new languages. Currently, there has been some work on the idea of multilinguality in ontologies such as EuroWordNet (Vossen 1998), bilingual WordNet, or BOW (Huang et al. 2010), and in the context of ontology localisation, such as Espinoza et al. (2008) and (2009), Cimiano et al. (2010), Fu et al. (2010) and Navigli and Penzetto (2010). Current work in machine translation has shown that word sense disambiguation can play an important role by using the surrounding words as context to disambiguate terms (Carpuat and Wu 2007; Apidianaki 2009).

One of the most interesting hypothesis is the one underlying interlingua RBMT systems. It uses an abstract intermediate semantic/logical representation to be used for translating into any target language. This hypothesis is converted into a SMT-viable alternative in which predicate-argument structures of both source and target language bitexts are used to bootstrap the SMT alignment module. This is what Wu and Palmer (2011) propose with the aim to abstract away from language specific syntactic variation and provide a more robust, semantically coherent alignment across sentences. As

the authors comment, a number of previous attempts had been made to either align deep syntactic/semantic lemmatized representations (as Marecek 2009a, b) did for English/Czech parallel corpus alignment); or to introduce semantic roles and syntax based argument similarity to project English Framenet to German, where however only the source was annotated. Choi et al. (2009) and Wu et al. (2010) enhanced Chinese-English verb alignments using parallel PropBanks. However there was no explicit argument mapping between the aligned predicate-argument structures.

HPBMT with Semantic Role Labeling

Recently there has been increased attention on using semantic information in machine translation. Pighin and Màrquez (2011) present a model for the inclusion of semantic role annotations in the framework of confidence estimation for machine translation. The model has several interesting properties, most notably: (1) it only requires a linguistic processor on the (generally well-formed) source side of the translation; (2) it does not directly rely on properties of the translation model (hence, it can be applied beyond phrase-based systems). These features make it potentially appealing for system ranking, translation re-ranking and user feedback evaluation. Preliminary experiments in pairwise hypothesis ranking on five confidence estimation benchmarks show that the model has the potential to capture salient aspects of translation quality.

Liu and Gildea (2008, 2010) proposed using Semantic Role Labels (SRL) in their tree-to-string machine translation system and demonstrated improvement over conventional tree-to-string methods. Wu and Fung (2009) developed a framework to reorder the output using information from both the source and the target SRL labels, and their approach uses the target side SRL information in addition to a Hierarchical Phrase-based Machine Translation framework. The proposed method extracts initial phrases with two different heuristics. The first heuristic is used to extract rules that have a general left-hand-side (LHS) non-terminal tag X, i.e., Hiero rules. The second will extract phrases that contain information of SRL structures. The predicate and arguments that the phrase covers will be represented in the LHS non-terminal tags. After that, they obtain rules from the initial phrases in the same way as the Hiero extraction algorithm, which replaces nesting phrases with their corresponding non-terminals. By applying this scheme, rules will contain SRL information, without sacrificing the coverage of rules. Such rules are called SRL-aware SCFG rules. During decoding, both the conventional Hiero-style SCFG rules with general tag X and SRL-aware SCFG rules are used in a synchronous Chart Parsing algorithm. Special conversion rules are introduced to ensure that whenever SRL-aware SCFG rules are used in the derivation, a complete predicate-argument structure is built. Gao and Vogel (2011) propose of using Semantic Role Labels to assist hierarchical phrase-based MT. They present a novel approach of utilizing Semantic Role Labeling (SRL) information to improve Hierarchical Phrase-based Machine Translation, by proposing an algorithm to extract SRL-aware Synchronous Context-Free Grammar (SCFG) rules. Conventional Hiero-style SCFG rules are extracted in the same

framework. Special conversion rules are applied to ensure that when SRL-aware SCFG rules are used in derivation, the decoder only generates hypotheses with complete semantic structures. They then perform machine translation experiments using nine different Chinese-English test-sets. The approach achieved an average BLEU score improvement of 0.49 as well as 1.21 point reduction in TER.

When dealing with formalisms such as semantic role labeling, the coverage problem is also critical., so it is important to follow Chiang's (2007) observation to use SRL labels to augment the extraction of SCFG rules. The formalism provides additional information and more rules instead of restrictions that remove existing rules. This preserves the coverage of rules.

Multiword Units in Dependency Structure

In Hwidong Na and Jong-Hyeok Lee (2011), another important contribution comes from the use of multiword units which had already been proposed in the computer assisted MT scenario by Wehrli et al. (2009). Here on the contrary, the translation requires non-isomorphic transformation from the source to the target. However, learning multi-word units (MWUs) can reduce non-isomorphism. They present a novel way of representing sentence structure based on MWUs, which are not necessarily continuous word sequences. The proposed method builds a simpler structure of MWUs than words using words as vertices of a dependency structure. Unlike previous studies, they collect many alternative structures in a packed forest. As an application of the proposed method, they extract translation rules in the form of a source MWU-forest to the target string, and verify the rule coverage empirically. On the same subject see also Carpuat and Diab 2010; Lambert and Banchs 2005; Ren et al. 2009.

Ontologies and Taxonomies

McCrae et al. (2011) widely use ontologies and taxonomies to organize concepts providing the basis for activities such as indexing and as background knowledge for NLP tasks. As such, translation of these resources would prove useful to adapt these systems to new languages. However, they show that the nature of these resources is significantly different from the "free-text" paradigm used to train most statistical machine translation systems. In particular, significant differences in the linguistic nature of these resources can be seen and such resources have rich additional semantics. As a result of these linguistic differences, standard SMT methods, in particular evaluation metrics, can produce poor performance. Leveraging these semantics for translation can be approach in three ways: by adapting the translation system to the domain of the resource; by examining if semantics can help to predict the syntactic structure used in translation; and by evaluating if existing translated taxonomies can be used to disambiguate translations. Results from these experiments shed light on

the degree of success that may be achieved with each approach. Rather than looking for exact or partial translations in other similar resources such as bilingual lexica, in the paper an adequate translation is presented using statistical machine translation approaches that also utilise the semantic information beyond the label or term describing the concept, that is relations among the concepts in the ontology, as well as the attributes or properties that describe concepts.

Latent Semantic Indexing

It is evident that the main-stream statistical machine translation is unable to tackle source-context information in a reliable way has been already recognized as a major drawback of the statistical approach, whereas (Carl and Way 2003) have proven the use of source-context information has been proven to be effective in the case of example-based machine translation. In this regard, (Carpuat and Wu 2007, 2008; Haque et al. 2009; España-Bonet et al. 2009; Banchs and Costa-jussà 2010) have already reported attempts to incorporate source-context information into the phrase-based machine translation framework. However, no transcendental improvements in performance have been achieved or, at least, reported yet.

Rafael E. Banchs & M.R. Costa-jussà (2011) proposed and evaluated an approach that uses a semantic feature for statistical machine translation, based on Latent Semantic Indexing. The objective of the proposed feature is to account for the degree of similarity between a given input sentence and each individual sentence in the training dataset. This similarity is computed in a reduced vector-space constructed by means of the Latent Semantic Indexing decomposition. The computed similarity values are used as an additional feature in the log-linear model combination approach to statistical machine translation. In the implementation, the proposed feature is dynamically adjusted for each translation unit in the translation table according to the current input sentence to be translated. This model aims to favor those translation units that were extracted from training sentences that are semantically related to the current input sentence being translated. Experimental results on a Spanish-to-English translation task on the Bible corpus demonstrate a significant improvement on translation quality with respect to a baseline system.

Crucial semantic problems are dealt with in a recent paper by Baker et al. 2012, on semantic issues like modality and negation which are relevant for SMT or what they call Semantically Informed Syntactic MT.

Evaluation Methods and Tools

To comment on this topic I will refer to a paper by Forcada et al. (2011b) – but see also Daelemans and Hoste 2009 – who extensively presents and experiments with evaluation metrics. One of the most widely used automatic MT evaluation metrics

is BLEU; then we have the NIST evaluation metric, the GTM metric based on precision and recall. Some of the metrics presented are language specific, and they are: METEOR, METEOR-NEXT, TER-plus and DCU-LFG. These metrics need specific resources and tools which are at present only available for English – see Kirchhoff et al. 2007 for semi-automatic evaluation.

BLEU (Papineni et al. 2002) – the most widely used automatic MT evaluation metrics – is a string-based metric which has come to represent something of a de facto standard in the last few years. This is not surprising given that today most MT research and development efforts are concentrated on statistical approaches; BLEU's critics argue that it tends to favour statistical systems over rule-based ones (Callison-Burch et al. 2006). Using BLEU is fast and intuitive, but while this metric has been shown to produce good correlations with human judgment at the document level (Papineni et al. 2002), especially when a large number of reference translations are available, correlation at sentence level is generally low. BLEU measures n-gram precision and the score is between 0 and 1. N-grams considered are any linear sequence of words up to length 4 (BLEU4), to be found in actual output and in reference translation. There is a brevity penalty in that single word match is just not counted if it never appears alone, like for instance the word "the"; and the same portion of text can't be used. BLEU is not sensitive to global syntactic structure; it doesn't care if the wrong translation is a function word rather than a content words or a proper name (input source sentences are all lower-cased and upper-case words are no longer visible). Human translation scored by BLEU typically falls around 60% – rather than 100% due to translator variations – and the best Chinese or Arabic translations into English may reach the same value (as reported in Ravi and Knight 2010).

The NIST evaluation metric (Doddington 2002) is also string-based, and gives more weight in the evaluation to less frequent n-grams. While this metric has a strong bias in favour of statistical systems, it provides better adequacy correlation than BLEU (Callison-Burch et al. 2006).

The GTM metric (Turian et al. 2003) is based on standard measures adopted in other NLP applications (precision, recall and F-measure), which makes its use rather straightforward for NLP practitioners. It focuses on unigrams and rewards sequences of correct unigrams, applying moderate penalties for incorrect word order.

METEOR (Banerjee and Lavie 2005; Lavie and Agarwal 2007) uses stemming and synonymy relations to provide a more fine-grained evaluation at the lexical level, which reduces its bias towards statistical systems. One drawback of this metric is that it is language-dependent since it requires a stemmer and WordNet.3. It can currently be applied in full only to English, and partly to French, Spanish and Czech, due to the limited availability of synonymy and paraphrase modules. METEOR-NEXT (Denkowski and Lavie 2010) is an updated version of the same metric.

The TER metric (Snover et al. 2006) adopts a different approach, in that it computes the number of substitutions, insertions, deletions and shifts that are required to modify the output translation so that it completely matches the reference translation(s). Its results are affected less by the number of reference translations than is the case for BLEU. Also, the rationale behind this evaluation metric is quite simple to understand for people who are not MT experts, as it provides an estima-

tion of the amount of post-editing effort needed by an end-user. Another metric based on error rates which preceded TER is WER (Nießen et al. 2000). WER and its extension mWER (Nießen et al. 2000) have been omitted from the experiments reported here as they seem to have been superseded by more recent metrics.

TER-plus (Snover et al. 2009) is an extension of TER using phrasal substitutions relying on automatically generated paraphrases, stemming, synonyms and relaxed shifting constraints. This metric is language-dependent and requires WordNet. It has been shown to have the highest average rank in terms of Pearson and Spearman correlation (Przybocki et al. 2008).

The DCU-LFG metric (Owczarzak et al. 2007) exploits LFG dependencies and has only a moderate bias towards statistical systems. It requires a dependency parser. Xiong et al. 2010b use linguistic feature to detect errors.

It should be noted that among the above measures, METEOR, METEOR-NEXT, TER-plus and DCU-LFG can only be used for English as a target language at the present time, given the language-specific resources that they require.

A Translation Example: Comparisons and Comments

Just to show how syntax, morphology and semantics may play an important role, we will use an example (Wilks/Zampolli 1994:592) from one of the many papers that Yorick Wilks has published on the subject (see Wilks 2009). The example is interesting in that it introduces the need to take care of agreement in discontinuous constituents. We will use common online translation systems (a RBMT Systrans and a SMT Google), will compare translations into three common European languages, and will comment on the errors produced:

(1) The soldiers fired at the women and I saw several fall
Google: (Ita) I soldati hanno sparato alle donne e ho visto cadere molti
(Fre) Les soldats ont tiré sur les femmes et j'ai vu la chute de plusieurs
(Germ) Die Soldaten schossen auf die Frauen, und ich sah mehrere Sturz
---> I soldati hanno sparato contro le donne, e ho visto cadere molti
Systran: (Ita) I soldati fatti fuoco contro le donne e me hanno veduto parecchio caduta
(Fre) Les soldats mis le feu aux femmes et à moi ont vu plusieurs chute
(Germ) Die Soldaten, die an den Frauen und an mir gefeuert wurden, sahen einiges Fall

On the whole, Google produces acceptable translations even though agreement is wrong both in Italian and French. In addition, German translation introduces a noun instead of the infinitival – Sturz translates the base verb form "fall" treating it as a noun. The treatment of the complement of "fired" (at the women) are all fine in the three languages, which we certainly regard as an achievement possible thanks to statistics: both the preposition and case are fine. If we look at Systran's translation on the contrary, we see an attempt to control gender in Italian: "caduta" is a feminine singular, but "parecchio" is masculine singular. So it would seem that there is no provision for the treatment of Number (both should have been plural). Another mistake is the presence of "me" in front of "hanno veduto", which is not only wrong –

"vedere"/see is not like "piacere", a psych verb that turns the deep experiencer subject into a dative. In fact "me" is accusative and as such it cannot be used in the subject position of any verb unless it is followed by another clitic that is in the accusative form as in "me lo"/to me it. The mistake is clearly due to the wrong phrase produced by joining "at the women and I" as if they were bound to the same preposition "at".

The auxiliary form is right but the past participle "veduto" is an archaic or stylistically marked version of "visto" that translates the simple past of "see". The question is that the main clause does not have a tensed main verb anymore: "fatti" is an absolute participial clause and translates "fired" as a past participle and not as a simple past. This ambiguity is quite common in English: in fact almost all verbs – with the exception of those irregular forms that are different in the two tenses, like "went/gone" – are ambiguous and require a disambiguation procedure in the parser to tell one tense from the other. The mistake is clearly related to the lack of statistical measures associated to the choice. Also French is semantically wrong: "mis le feu" does not really fit into the translation required here, it translates the other meaning of "fire", BURN. It is difficult to understand the semantic choice here, because you don't currently BURN WOMEN very easily, even though that might have happened in the past. Actually in the Middle-Ages when witches were around, many women were set on fire on a pyre. As to the conjoined sentence, we see again the same mistake of using a dative "à moi" in French and "an mir" in German, rather than simply introducing a nominative pronoun, like "moi" and "ich". Then the quantifier "several" is again translated without agreement in both French and German: however French "plusieurs" captures Number and the German "einiges" is wrong both in agreement and in meaning – it translates "some". In fact, the German translation is primarily wrong because it turns the two conjoined sentences as if they were headless relative clauses – which is possible in English but not in German – parsing the constituents "fired at the women and I" as if they were a well-formed structure. This is apparent from the introduction of two commas, at the beginning and at the end of the conjunct "…, die an den Frauen und an mir gefeuert wurden," and by the use of passive which is clearly nonsensical given the presence of a nominative Agent "die Soldaten".

So eventually, Systran has produced a far worse result than Google, which by making use of its enormous terabyte of parallel texts, has shown the power of SMT. More examples follow below.

MT for the Future

New Statistical Methods and a Comprehensive Translation Model

We assume that the right direction for MT of the future is to incorporate both syntax and semantics in its statistics. We can do this in these two ways:

A. A first way would be the one proposed by the LOGON project (Oepen et al. 2004, 2005; Oepen and Lønning 2006). It increases the role of NLP tools and

leaves mainly to statistics the final re-ranking of best translation candidates, as will be better explained below. This is how the authors summarize their approach: "a hybrid MT architecture, combining state-of-the-art linguistic processing with advanced stochastic techniques. Grounded in a theoretical reflection on the division of labor between rule-based and probabilistic elements in the MT task … combining component-internal scores and a number of additional sources of (probabilistic) information, … explore discriminative re-ranking of n-best lists of candidate translations through an eclectic combination of knowledge sources";

B. A second way is the one Tan et al. 2012 propose, in their seminal work. They present a new language model which is an "a large scale distributed composite language model that is formed by seamlessly integrating n-gram, structured language model and probabilistic latent semantic analysis under a directed Markov random field paradigm to simultaneously account for local word lexical information, mid-range sentence syntactic structure, and long-span document semantic content". That is, they try to combine semantics/pragmatics, syntax and string-based statistical processing

As for method B., in the abstract to their article they present their approach as follows:

> The composite language model has been trained by performing a convergent N-best list approximate EM algorithm and a follow-up EM algorithm to improve word prediction power on corpora with up to a billion tokens and stored on a supercomputer. The large scale distributed composite language model gives drastic perplexity reduction over n-grams and achieves significantly better translation quality measured by the BLEU score and "readability" of translations when applied to the task of re-ranking the N-best list from a state-of-the-art parsing-based machine translation system. (ibid., 1)

The reason to resort to such an approach reflects the obvious fact that – as the authors note-, the technology based on n-grams has reached a plateau and there is a desperate need to find a new approach to language modeling (Lavie et al. 2006). Work on Chinese has pushed the over of n-gram up to 6-gram obtaining better translations, but the improvement beyond that is minimal (Zhang 2008).

Wang et al. (2006) studied the stochastic properties for a composite language model that integrates n-gram, probabilistic dependency structure in structured language model (SLM), and probabilistic latent semantic analysis (PLSA) under the directed Markov random fields (MRF) framework (Wang et al. 2005). They derived another generalized inside-outside algorithm to train composite n-gram, SLM and PLSA language model from a general EM algorithm by following Jelinek's ingenious definition of the inside and outside probabilities for SLM (Jelinek 2004).

Eventually, the authors are aiming to influence word prediction to find best word pair triggers, with both (dependency) syntactic and (discourse topic) semantic/pragmatic information (Wallach 2006). The output can be combined and used with trigrams in a composite model to drive the final decoder. The resulting language model is defined as the "composite 5-gram/2-SLM+2-gram/4-SLM+5-gram/PLSA1 language model". The interesting part of the evaluation, which as expected is reported

to increase BLEU scores by a 1.19%, is the one dedicated to the "readability" of Chinese-English translations, where they ask human judges to evaluate semantic versus grammatical correctness. In a table the authors report the results of "readability" evaluation on 919 translated sentences of 100 documents, and divide the sentences into four groups: perfect, only semantically correct, only grammatically correct, wrong. The evaluation shows improvements when going from baseline and simple string-based 5-gram processing to the composite language model created by the authors. The greatest relative variation is shown by G(rammatical) sentences, over 60% increase; then P(erfect) sentences, over 50% increase; the lowest relative variation is in S(emantic) sentences that only increase by a 7%. Overall totally wrong sentences decrease by 25%, again a remarkable achievement. This notwithstanding, we can easily see that the amount of "readable" sentences reaches the 66% of the total 919 from a starting point of 56%. However, it is important to stress that they obtain these results by training on a 1.3 billion word corpus using a supercomputer and will not be duplicated on smaller hardware in the near future.

Interesting enough, the results above are almost comparable to those obtained by the LOGON project, in which with a totally different technology and a much smaller corpus, they carried out an evaluation on domain-bounded sentences of unseen running text, they found that on the two thirds (62%) that have been translated, they reached an accuracy of 72.28%. The evaluation carried out by Johannessen et al. 2008 with the help of human judges, based on quality parameters such as "fidelity" and "fluency" showed a result of around 2 points on a graded scale that goes from 0 to 3, where 2 is translated as fair fidelity and still some mistakes in fluency. As the authors themselves comment, the scarsity of resources existing for Norwegian has been the main motivation for building a semantic-transfer-oriented translation system that uses stochastic processing for the target language model, English (see also Llitjós and Vogel 2007). Minimal Recursion Semantics is the "glue" which performs transfer from source to target language and serves as the information vehicle between LFG and HPSG. For a similar approach purely cast in LFG see Riezel and Maxwell 2006. Purely statistical approaches are doomed to failure. In addition the probabilistic NLP experience by itself suggests that a "ceiling" effect has already been reached. As the authors say,

> The Norwegian LOGON initiative capitalizes on linguistic precision for high-quality translation and, accordingly, puts scalable, general-purpose linguistic resources—complemented with advanced stochastic components—at its core. Despite frequent cycles of overly high hopes and subsequent disillusionment, MT in our view is the type of application that may demand knowledge-heavy, 'deep' approaches to NLP for its ultimate, long-term success. (ibid., 144)

Eventually (Bellegarda 2001, 2003) anticipated what is needed, that is a "more polyvalent, multi-faceted, effective and tractable solutions for language modeling – this is only beginning to scratch the surface in developing systems capable of deep understanding of natural language" . In order to achieve this, it is not sufficient to increase the size of data to obtain a breakthrough in the performance. It has been shown that it is not the complicacy of the algorithm that makes the difference: simple algorithms may outperform more complicate ones. However, as Tan et al. have demonstrated, increasing the size of the data has brought improvements, but

then it has been the increase in the complexity of the model that has made the difference. "For language modeling in particular, since the expressive power of simple n-grams is rather limited, it is worthwhile to exploit latent semantic information and syntactic structure that constrain the generation of natural language, this usually involves designing sophisticated algorithms. Of course, this implies that it takes a huge amount of resources to perform the computation." (ibid., 49) Of course, for this to become a feasible alternative, a large scale distributed language model would be required, possibly via cloud computing. Their conclusions are as follows, they intend to

> … construct a family of large scale distributed composite lexical, syntactic, and semantic language models. Finally we'll put this family of composite language models into a phrased-based machine translation decoder that produces a lattice of alternative translations/transcriptions or a syntax-based decoder that produces a forest of alternatives (such integration would, in the exact case, reside in an extremely difficult complexity class, probably PSPACE-complete) to significantly improve the performance of the state-of-the-art machine translation systems.

What really matters at this point is confirming the increasing improvements of SMT while at the same time keeping strictly in mind its inherent limitations. Martin Kay, in one of his latest papers (2011), nicely expresses his pessimism and at the same time optimism towards possible future prospects of MT. He connects MT to what a human translator is doing when translating between English and French, coming to the obvious conclusion that the output is inextricably bound to cultural issues and not just a matter of lexical, semantic or structural knowledge. In fact, by just increasing the size of the training data, one might come up with more cultural problems to solve. Further, the language model would be less adequate the more data are introduced, unless they belong strictly to the same domain, or as he puts it "… new data generally opens at least as many questions as it settles" (ibid., 18). The problem is that "… there is much in any but the most trivial texts that a reader must infer from what is made explicit, but what is explicit in a translation in another language is not generally the same, so that substantive information is both added and subtracted by the translator"(ibid., 15). I will not report his examples, but we can all can try examples on any online available translator to realize the truth of his statement. In fact, what is implicit, is not just "pragmatically" based and culturally motivated, but also in some cases, specifically language related.

We already saw one example above – from English to agreement aware languages like German, French and Italian-, in which researchers discussed questions related to agreement. Now I will turn the other way around. Implicit elements in most cases correspond to "Empty" or "Null" elements as they are usually defined. For instance, in Penn Treebank, they have been manually classified and counted, and the total number for an English treebank is over 36,862 cases of null elements (including traces, expletives, gapping and ambiguity) as listed in Johansson and Nugues (2007), in other words, one every complex sentence, and one every three simple sentences. Then there is the problem of coindexation, or assignment of an antecedent to the empty element: 8,416 are not coindexed, that is 22.83% (see Dienes and Dubey 2003; Schmid 2006). If we exclude all traces of WH and topicalization and limit ourselves to the category OTHER TRACES which includes all unexpressed SBJ of infinitivals

and gerundives, we come up with 12,172 cases of Null non-coindexed elements, 33% of all cases. However, these numbers are this is still a small percentage when compared to languages like Chinese (Cai et al. 2011; Yang and Xue 2010) or some Romance languages like Italian which allow for free null subjects (also objects in Chinese) insertion in tensed clauses. In our treebank of Italian called VIT (Tonelli et al. 2008; Delmonte et al. 2007), we counted an addition of 51.3% of simple sentences with non-canonical, or lexically unexpressed subjects. Obviously this covers the total number utterances in the small corpus (60K tokens) of transcribed spoken dialogues, where the implicit is much higher than in the written text.

Here below are some example translations from Italian to English – but I assume they could easily be from Portuguese, but also from Japanese, and Chinese to English – in which we quite simply demonstrated that the "implicit" (Delmonte 2009a, b), might in many cases determine what is missing in the translation and is, in fact, desperately relevant. Computing complete Predicate-Argument structures is essential for Machine Translation tasks – as Chung and Gildea (2010) have shown where one of the two languages belongs to typology above. As an example, we tried the translation of one sentence from Italian into English, introducing null elements and lexical pronouns, both on Systran and Google online translation websites:

Italian Original

Maria successivamente, dopo aver rifiutato la sua offerta, gli ha detto che vuole vendere la propria casa a sua sorella perché vuole aiutarla.

Gold Translation

Then, after having rejected his offer, Maria told him that she intends to sell her (own) house to her sister because she wants to help her.

Google Translation

Maria later, after she refused his offer, told him he wants to sell his house to his sister because she wants to help.

Systran Translation

Maria successively, after to have refused its offer, she has said it that she wants to sell own house to its sister because she wants to help.

The sentence is fairly simple both in lexical choice and syntactic structure. As one can be gather, Google makes grammatical mistakes due to lack of long distance control – "he, his, his" are all in masculine gender rather than feminine. Systran gets the subject empty pronouns right, but then mistakes the possessives – "its" is neutral – and uses infrequent adverbials like "successively" to translate "dopo". As usual, Google gets an overall best translation both for the grammatical and lexical aspects. Neither of the translation includes the object enclitic "-la"/her in the output. In fact, the verb "help" can be used intransitively, i.e. omitting the object and no mistake ensues. However in this way the leftover pronoun is implicit and needs to be evoked. If we substitute "aiutarla" with "lasciarla" we obtain two different behaviours. Google produces the same output: no pronoun. In this case, however the meaning is no longer preserved and "she wants to leave" has a totally different meaning from

"she wants to leave her". Systran on the contrary produces "it" for singular no matter what gender it is ("lo", "la"), and "them" for plural.

We will take again Kay's preliminary conclusion on the topic to close the section,

> Since examples of the kind just considered are clearly beyond the reach of current, or any readily foreseeable technology — especially if based on machine learning — we must take it that they do nothing but degrade the best performance of the systems that are learned from the texts that contain them. Supervised learning from a corpus of translations that were stricter, if less idiomatic, should surely be expected to result in superior systems. But large corpora of such translations do not occur naturally, would be expensive to produce artificially, and would be required to meet different criteria as the field progressed.

Speech-to-Speech MT

This section fully addresses what, in my opinion, will be the driving application for the future of MT, the one that most users will come to terms with, using mobile devices and other similar technologies. At the heart of Speech-To-Speech MT or S2S for short, there is the need to communicate worldwide orally, for many different kinds of purposes. In other words, we are talking about the need to implement systems for multimodal multilingual communication. This is the future of man–machine interface programs and at the heart of any future development in the associated fields not only of Artificial Intelligence, Speech Synthesis and Automatic Speech Recognition, but also Image Processing, Computer Vision and Face Recognition to be used also in robotics. Thus, MT is only one facet of this important application domain that is based on advancements in basic fields of research like computational linguistics, pattern recognition, and machine learning. Multilingual tools of the future will have to incorporate some if not all of these facilities in order to make real breakthrough in the application market. It is quite obvious to me that a multilingual translation system that is able to take advantage of both spoken and visual input is by far more promising that its companion system that only makes use of written input in a dialogue situation (Zong et al. 2002).

S2S end-to-end systems are organized in a number of complementary modules that receive spoken input in one language and elaborate the corresponding spoken form in another (Karakos and Khudanpur 2008). This is one possible pipeline:

- Speaker produces an utterance in language A, to a device that is linked to an ASR system
- The ASR turns the spoken utterance in its transcribed version still in language A
- A system of utterance and dialogue understanding computes its meaning via an NLP system – this would be the interlingua based approach
- The interlingua is then passed to a Language Generator for language B.
- OR
- The sentence is passed to the MT system that produces the most probable translation into a language B by choosing the best candidate in a list
- The translated sentence is passed to a Speech Synthesizer for language B which speaks it into a device for the user speaker of language B.

Zhang Ying (Joy) (2004) reports in his Survey of Current Speech Translation Research that the translation engines utilized were basically a Multi-Engine MT (MEMT) system,

> … whose primary engines were an Example-Based MT (EBMT) engine and a bilingual dictionary/glossary. Carnegie Mellon's EBMT system uses a "shallower" approach than many other EBMT systems; examples to be used are selected based on string matching and inflectional and other heuristics, with no deep structural analysis. The MEMT architecture uses a trigram language model of the output language to select among competing partial translations produced by several engines. It is used in this system primarily to select among competing (and possibly overlapping) EBMT translation hypotheses. The translated chaplain dialogs provided some of the training. Pre-existing parallel English-Croatian corpora is also used. An addition finite-state word reordering mechanism was added to improve placement of clitics in Croatian. (ibid., 2–3)

Systems like SPEECHLATOR (Waibel et al. 2003), or MASTOR (Gao et al. 2008) work in interlingua modality. All applications have been cast in limited domains and in particular in the hotel room reservation task and have to cope with spontaneous speech. Most importantly, the C-Star consortium, Consortium for Speech Translation Advanced Research, used interlingua, defined as "a computer readable intermediate language that describes the intention of a spoken utterance of a particular language" (ibid., 5), and the domain was related to travel planning. Translating the "intention" allowed system designers to substitute sloppy ramblings in the input or different ways of expressing the same meaning with the one translation available. ***Nespole!*** (Lavie et al. 2002) was one such system which followed another similar system called JANUS III (Levin et al. 2000). Other interesting systems were Digital Olympics (Zong 2008) produced for the Olympics in Peking, and cofunded by the European authorities and the Chinese government, and the NEC Speech Translation System (Yamabana et al. 2003).

The second case of translation is referred to systems like ATROS (Automatic Trainable Recognizer of Speech) developed in the EuTrans project, which aim to synchronize speech recognition models with linguistic levels like lexical, syntactic and eventually translation model, by the use of FST. The MT technique followed is example-based. The AT&T approach uses multimodal parsing and understanding always with a finite-state model. The system subdivides the translation task into a phase for lexical choice and another phase for lexical reordering (ibid., 5). The lexical choice phase is divided up into phrase-level and sentence-level using different translation models. Eventually, the reordering phase approximates a tree-transducer using a string transducer.

To describe current state-of-the-art STSMT we will be referring to the international challenge and associated evaluation campaign called QUAERO, reported in Lamel et al. 2011, for a bidirectional French-German task, which has seen the participation of the most important actors on the scene: RWTH, KIT, LIMSI and SYSTRAN. In the Reference section websites of the partners involved in the common task are reported.

Problems related to STSMT are common and different from standard written-based SMT. First of all, there is the need to make available a recognition vocabulary

big enough to include all word-forms possibly present in the task at hand in order to reduce the number of Out Of Vocabulary (OOV) rates (see Gales et al. 2007). This constitutes the worst problem to solve, and it is not just a matter of increasing a list, with frequency of occurrence. Words included in the recognition vocabulary needs to be represented phonetically. Vocabulary sizes range from 65K to 300K words as reported in Lamel et al. and OOV rates range from around 0.5% to 2%. It is interesting to note that systems represent the pronunciation dictionary with sets of phone symbols that go from 35 up to a maximum of 50 symbols. Systems generate the phonetic representation with different methods: some use rule based grapheme to phoneme conversion, others statistical methods, or a combination of the two, often introducing a list of manually verified exceptions. Most phone sets include pseudo phones for silence and non-speech sounds and there are typically 1.1–1.3 pronunciations per word. However, as will be explained below, there are special provisions for prosodically related pronunciation variants, which in a language like French, constitute a frequent phenomenon with which to cope.

In STSMT in addition to language models and translation models, there are acoustic models to build. In particular, current acoustic models are trained on several hundreds of hours of audio data coming from a variety of sources. Language models are trained on over a billion words of texts, comprised of assorted newspaper and newswire texts: they end up typically containing around 400M 4-grams for both languages.

It is important to note that all speech recognition experiments described in Lamel et al. were performed with the help of the Janus Recognition Toolkit (JRTk) from CMU and the Ibis single pass decoder described in Soltau et al. 2001. The Janus system is described in (Lavie and Waibel 1996; Levin et al. 2000). Training for German was performed using some 350 h of training material from different sources. As reported in Lamel et al.

> Two different front-ends were applied: The warped minimum variance distortionless response (WMVDR) approach and the conventional (Mel-frequency Cepstral Coefficients) MFCC approach. The front-end uses a 42-dimensional feature space with linear discriminant analysis and a global semi-tied covariance (STC) transform with utterance-based cepstral mean and variance normalization. The 42-dimensional feature space is based on 20 cepstral coefficients for the MVDR system and on 13 cepstral coefficients for the MFCC system … All the acoustic data is in 16 kHz, 16 bit quality. Acoustic model training was performed with fixed state alignments and Vocal Tract Length Normalization (VTLN) factors … The system uses left-to-right hidden Markov Models (HMM)s without state skipping with three HMM states per phoneme.

This produced an adapted gender- and speaker-independent acoustic model. The Language Model for German was built from a variety of text sources and resulted in a 10GB LM, containing 31.7M 2-grams, 91.9M 3-grams, 160.4M 4-grams, as reported in the same paper. Speaker adaptation was performed in a second pass and produced FSA-SAT models with language models which were even bigger.

For French, approximately 330 h of audio data were used to train the acoustic models. The segmentation was implemented in two steps applying an HMM-based segmenter which took into consideration different speech events, noises, silences and music. For each speech segment a Gaussian Mixture Model is generated. Two different

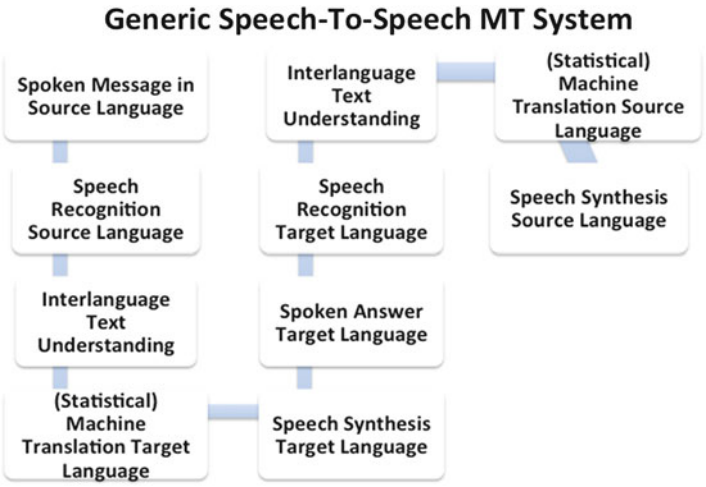

Fig. 6.3 Generic pipeline for a STSMT system

kinds of phoneme sets are used for training: a first one consisting of 35 phonemes and another version provided by Vocapia that consists of 32 phonemes. As with German. two types of acoustic front-end were used: one based on Mel-frequency Cepstral Coefficients and the other one on the warped minimum variance distortionless response. Both front-ends work on a window of 10ms. At the end of the training process five acoustic models were produced, which were improved by boosted Maximum Mutual Information Estimation training. LMs were trained with all tests available using the SRI Language Modeling Toolkit. Interesting enough, for the training procedure a dictionary was used which contained hesitations, fragments, human and non-human noise in addition to pronunciation variants of each word, taken from GlobalPhone and Lexique 3. Missing pronunciations were generated with Sequitur G2P (Bisani and Ney 2008). The final system reached 27% WER on the Quaero Development Set. As reported in the Conclusions, there has been a steady progress in reducing the WER in the last 3 years. For some languages reduction reaches 25%, and it is around 15% for the three primary languages – French, German English (Fig. 6.3).

TED Conferences Talks and the WIT3 Corpus

TED Conference at TED (Technology, Entertainment and Design) website, www.ted.com, have been posting video recording of talks, having as their subject cultural issues in general. Talks come with English subtitles and their translations in more than 80 languages. This has been done since 2007 for the sake of sharing ideas around the world, as the organizer comment on the website.

FBK (Bruno Kessler Foundation) in Trento (Italy) have organized a website https://wit3.fbk.eu/, with the aim of redistributing the whole corpus with original textual

contents in their multilingual transcriptions, but also to make a ready-to-use version with MT benchmarks and processing tools for research purposes. The acronym of the website stands for Web Inventory of Transcribed and Translated Talks. A detailed description of the corpus can be found in a recent paper by Cettolo et al. 2012.

The same organizers, including Marcello Federico, Mauro Cettolo together with Michael Paul from NICT (Japan) and Sebastian Stueker (KIT) Germany, are responsible for the TED Task Evaluation Campaign which is an important event related to IWSLT conferences, and can be found at http://iwslt2012.org/, subdirectory "evaluation-campaign/ted-task". TED Task includes the following subtasks:

> IWSLT proposes challenging research tasks and an open experimental infrastructure for the scientific community working on spoken and written language translation. The IWSLT 2012 Evaluation Campaign includes the TED Task, that is the translation of TED Talks, a collection of public speeches on a variety of topics. Three tracks are proposed addressing different research tasks:
>
> ASR track : automatic transcription of talks from audio to text (in English)
>
> SLT track: speech translation of talks from audio (or ASR output) to text (from English to French)
>
> MT track : text translation of talks for two language pairs plus eight optional language pairs:
>
> official: from English to French and from Arabic to English
>
> optional: from German, Dutch, Polish, Portuguese-Brazil, Romanian, Russian, Turkish and Chinese to English
>
> Main challenges of the proposed tracks are:
>
> Open domain ASR, clean transcription of spontaneous speech, detection and removal of non-words, and talk style and topic adaptation.
>
> Open domain SLT, translation of speech or ASR output into true-case punctuated text, and talk style and topic adaptation.
>
> Open domain MT between distant languages, and talk style and topic adaptation.

Full guidelines can be found on the same website. This is certainly not the only initiative – KIT, Germany would be another one – but certainly one of the most important ones.

The GALE DARPA MT Project

As discussed at the beginning of this chapter, a number of different institutions are currently contributing to finance research efforts of the vast community of scientists working in the field of MT. The most important of these international initiatives in favour of the improvement of MT research is the one financed by DARPA. The project was originally called GALE and has recently partially concentrated on BOLT (Broad Operational Language Translation), which has clear military goals. As to this project, the interesting thing that happened last year, was the hiring of Professor Bonnie Dorr as manager of BOLT: this event has an extremely important meaning. It testifies to the switch of perspective the DARPA management wants to give to the project: from statistics only, to the massive introduction of linguistics, that is syntax and semantics, into MT. Of course we regard this change of point of view a successful move towards finding the best approach for the MT of the future.

To comment on the GALE (Global Autonomous Language Exploitation) program we will be referring basically to the original webpage dedicated to the project directly on the DARPA website and from a presentation downloadable from their general internal search engine. DARPA has been a key sponsor of machine translation, as well as computer processing of language (speech and text) work for over three decades. GALE began in September 2005 and is still continuing even though it was scheduled to run only until September 2010. GALE speech-to-text and machine-translation researchers came from the following corporations and organizations, as reported by The Associated Press, in a 2006 online article on the topic: IBM Corp., backed by a $6 billion annual research budget; SRI International, a $300 million, nonprofit research organization based in Silicon Valley; and BBN Technologies Inc., a $200 million research contractor headquartered in Cambridge BBN nabbed people at Cambridge University, the universities of Maryland and Southern California and a French lab, among others. IBM got researchers from Carnegie Mellon, Johns Hopkins, Brown University and Stanford, plus other researchers at the University of Maryland. SRI's links included European and Asian schools, Columbia University and the universities of California and Washington. The goal is to create technology that will automatically translate spoken or written words from foreign languages into grammatically correct English. GALE has an ambitious goal of reaching 95% accuracy without human mediation. The technology is moving toward allowing the translations to happen in real time. These goals are set forth in ambitious research efforts and according to DARPA "these efforts are poised to come close to achieving their goals in certain specific contexts with Modern Standard Arabic and Mandarin Chinese". The largest of these efforts was the 5-year, multimillion-dollar-per-year GALE program, which seeks real-time translation of Modern Standard Arabic and Chinese print, Web, news, and television feeds. The second program is the Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) program. TRANSTAC's goal is highly specific: a portable two-way speech translation system that enables an average soldier to communicate with a person who cannot speak English. NIST Machine Translation Evaluation for GALE: The Speech Group in the Information Technology Laboratory's Information Access Division at NIST is undertaking the development of an evaluation for machine translation (MT) using "edit-distance" as the evaluation metric as defined by the GALE program. The GALE program will evaluate MT in terms of the quality of the system translations. This will be accomplished by measuring the edit distance between a system output and a gold standard reference. The term "edit-distance" refers to the number of edits (modifications) that someone (human) needs to make to the output of a machine translation system such that the resulting text is fluent English and completely captures the meaning of the gold standard reference. In order to achieve its goal, the GALE program will have to develop and apply computer software technologies to analyze and interpret huge volumes of speech and text. Output information will have to be in easy-to-understand forms for military personnel and monolingual English-speaking analysts to use, in response to direct or implicit requests. GALE will consist of three major engines: Transcription, Translation and Distillation. The output of each engine will be limited

to English text. The distillation engine integrates information of interest to its user from multiple sources and documents. Military personnel will interact with the distillation engine via interfaces that could include various forms of human-machine dialogue (not necessarily in natural language).

According to the project description on the LDC website, the Linguistic Data Consortium supports the GALE Program by providing linguistic resources – data, annotations, tools, standards and best practices limited to Arabic – for system training, development and evaluation. The Translation process will take Arabic source text drawn from many different genres, both spoken and written, and translate it (hopefully) into fluent English while preserving all of the meaning present in the original Arabic text. Translation agencies will use their own best practices to produce high quality translations, according to specific guidelines so that all translations are guided by some common principles. Linguistic Data Consortium will also be providing post-editing of MT output in order to compute "edit distance" between machine translations and human gold standard translations. The post editor's role is to compare computer-translated texts against the same texts translated by humans. Working with one sentence at a time, the editor modifies the computer translation until its meaning is identical to the human translated sentence.

As indicated above, GALE's goal was to deliver, by 2010, software that can almost instantly translate Arabic and Mandarin Chinese with 90–95% accuracy. Fortunately for the GALE teams, they didn't have to be near 95% right away. In the first year, they were expected to translate Arabic and Mandarin speech with 65% accuracy; with text the goal was 75%. In an interview reported on SLATE, Mari Maeda, a DARPA manager who ran the program, says that, by the end, "TransTac achieved about 80% accuracy: enough to be interesting, but not enough to be useful." Considering state of the art MT field that was already to be regarded as a particularly difficult goal to achieve. DARPA estimated that the best systems could translate foreign news stories at 55% accuracy. But DARPA wanted translations not only from such controlled, well-articulated sources: in fact, it was to be considered as an open domain, unlimited vocabulary task. As reported in The Associated Press article "GALE incorporates man-on-the-street interviews and raucous colloquial chats on the Web. Background noise, dialects, accents, slang, short words … that most speakers don't bother to clearly enunciate – these are the stuff of nightmares for speech-recognition and machine-translation engineers". We have presented and discussed at length all of these "nightmares" in the chapter. The test – hours of audio and dozens of documents in Arabic and Mandarin – and the evaluation was done by counting the number of human edits that the sentences needed in order for them to have the correct meaning. Again, quoting from the TAP article, "the results largely met DARPA's demands of 75% accuracy for text translation and 65% for speech… The BBN-led team produced 75.3% accuracy with Arabic text, 75.2% in Chinese. It scored 69.4% in Arabic speech; 67.1% in Mandarin. IBM scored higher with Arabic text and SRI scored higher in Mandarin. The current successor of TransTac is called BOLT and Bonnie Dorr, program manager forBOLT, says that DARPA is now "very focused on moving beyond statistical models." The reason is that, as you throw more and more paral-

lel data at your algorithms, you "get diminishing returns. The payoff gets smaller, and you start to plateau with your results even if you increase the volume of training data." (again reported on SLATE online).

If DARPA is financing LDC to produce additional translation of Arabic, this is a clear sign of two symptoms:

- More training data are required
- The current results of translation systems are still unsatisfactory
- Of course, we assume that a final strategic improvement will not come until another additional piece of research is added to the list:
- Inventing new translation and language models that will incorporate semantics and pragmatics (besides syntax) in a most fruitful way, which is not yet the case
- Combining more systems in a pipeline to produce a hybrid hypersystem that can learn

This is clearly what the MT community as a whole is striving for and what I assume will happen in future.

Conclusion

In this chapter I have presented and commented what I regard the most interesting technologies and methodologies for Machine Translation, including both Rule-Based and Statistically-Based systems. I devoted a first section to purely statistically based systems, highlighting their shortcomings and their advantages which make them more and more important for the future of MT. In fact, a statistical model is shown to be an essential component of most Rule-Based systems reviewed in another section: these systems take advantage of the ability of generalization that statistics makes easily available to create what are usually called hybrid systems. A third section has been devoted to syntactically based systems which make use of syntactic tree structures of dependency structures to produce better modelling of the data and hopefully better translations. These systems are strongly dependent on computational linguistic tools like parsers, morphological analysers and suffer from their shortcomings which impinge on the final translation accuracy level.

I take graph-based models to be superior in general to phrase-based or word-based models, the reason being simply the fact that structural properties of both input and output can be duly taken into consideration in the modelling statistical phase. Why this is important should now be clear: in order to produce a real step forward, Machine Translation should incorporate properties belonging to both syntax and semantics of the sentence and text to be translated. This can only be achieved with a structurally aware statistical model. I take models relying on dependency structure to be the reference point with additional constraints however: the need to satisfy predicate-argument restrictions as realized in the statistical graph-based model and reinforced in the observed data.

Rule-Based systems could still be useful to encode Multiwords correspondences and other idiomatic expressions which require some preprocessing. However, as the case of feature agreement has clearly shown, Rules-Based systems are unable to enforce local matching requirements when fine-grained coupling is needed. They could perhaps work in postprocessing to check such local agreements with highly targeted rule systems, which must be language dependent.

Eventually, speech-to-speech multilingual processing has started to be used in real life applications. This is great news, but also bad news as far as current achievements are concerned. The effort however is enormous and the quantity of resources in play is outside the scope of any single researcher computing ability. Only specialized centers may legitimately aim at competing in such international challenges. What about the use of visual computing and the interpretation of facial movements or other additional gestures? Multimodal computation is still in its infancy and its interaction with natural language processing tools is expected to grow in the future. As to current situation, I don't know of any system capable of taking advantage of gestures of facial expressions to improve its multilingual tools.

New mathematical models are needed that incorporate all types of knowledge needed to come as close as possible to what human translators do: best translators are always domain constrained, and this applies to both humans and computers. Syntax poses different challenges from pragmatics and semantics: new mathematical models need to take these differences into adequate account.

Last but not least, the need to foster improvements in the companion field of computational linguistics, which alone can come up with complete linguistic representations needed in the Rule-Based scenario. I am referring to the need to enrich dependency structures with null elements and annotate them with coreference information to allow for proper agreement features to be instantiated. Such new tools could then be used to produce Logical Form representations to better handle meaning differences and ambiguities.

References

Abney S (1989) Parsing by chunks. In: Tenny C (ed) The MIT parsing volume, 1988–89. Center for Cognitive Science, MIT, Cambridge

Adly N, Alansary S (2009) Evaluation of Arabic machine translation system based on the universal networking language. In: The 14th international conference on applications of natural language to information systems "NLDB 2009", Saarland University, Saarbrucken, Germany, 23–26 June 2009

Alansary S, Nagi M, Adly N (2009) The universal networking language in action in English-Arabic machine translation. In: 9th conference on language engineering, Cairo, pp 1–12

Alegria I, Diaz de Ilarraza A, Labaka G, Lersundi M, Mayor A, Sarasola K (2007) Transfer-based MT from Spanish into Basque: reusability, standardization and open source, vol 4394, Lecture Notes in Computer Science. Springer, Berlin/New York, pp 374–384

Alexandrescu A, Kirchhoff K (2009) Graph-based learning for statistical machine translation, 2009. In: Human language technologies: the 2009 annual conference of the North American Chapter of the ACL, Boulder, Colorado, pp 119–127

Alkuhlani S, Habash N (2011) A corpus for modeling morpho-syntactic agreement in Arabic: gender, number and rationality. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics (ACL'11), Portland, Oregon, USA

Ambati V, Lavie A (2008) Improving syntax driven translation models by restructuring divergent and non-isomorphic parse tree structures. In: Proceedings of the eighth conference of the association for machine translation in the Americas, Waikiki, pp 235–244

Ambati V, Lavie A, Carbonell J (2009) Extraction of syntactic translation models from parallel data using syntax from source and target languages. In: MT Summit XII: proceedings of the twelfth machine translation summit, Ottawa, 26–30 Aug 2009, pp 190–197

Ambati V, Vogel S, Carbonell J (2011) Multi-strategy approaches to active learning for statistical machine translation. Associated press article on the web: http://www.cnn.com/2006/TECH/11/06/darpa.translation.ap/index.html

Apidianaki M (2009) Data-driven semantic analysis for multilingual WSD and lexical selection in translation. In: Proceedings of the 12th conference of the European chapter of the association for computational linguistics (EACL), Athens, 30 Mar–3 Apr 2009, pp 77–85

Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J, Davis A et al (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 25(1):25–29

Ayan NF, Zheng J, Wang W (2008) Improving alignments for better confusion networks for combining machine translation systems. In: Proceedings of the Coling'08, pp 33–40

Baker K, Bloodgood M, Dorr BJ, Callison-Burch, C, Filardo, NW, Piatko, C, Levin L, Miller S (2011) Modality and negation in SIMT – use of modality and negation in semantically-informed syntactic MT. Comput Linguist 38(2):1–48, (accepted for publication)

Baker K, Bloodgood M, Dorr BT, Callison-Burch C, Filardo NW, Piatko C, Levin L, Miller S (2012) Modality and negation in SIMT – use of modality and negation in semantically-informed syntactic MT. Comput Linguist 1:1–48

Banchs, RE, Costa-jussà MR (2010) A non-linear semantic mapping technique for cross-language sentence matching. In: Proceedings of the 7th international conference on advances in natural language processing (IceTAL), Reykjavik, pp 57–66

Banchs RE, Costa-jussà MR (2011) A semantic feature for statistical machine translation. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, ACL HLT 2011, Portland, June 2011, pp 126–134

Banerjee S, Lavie A (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Ann Arbor, June, pp 65–72

Banerjee P, Dandapat S, Forcaday ML, Groves D, Penkale S, Tinsley J, Way A (2011) Technical report: OpenMaTrEx, a free, open-source hybrid data-driven machine translation system

Béchara, H, Ma Y, van Genabith J (2011) Statistical post-editing for a statistical MT system. MT Summit XIII: the thirteenth machine translation summit [organized by the] Asia-Pacific association for machine translation (AAMT), Xiamen, 19–23 Sept 2011, pp 308–315

Bellegarda J (2001) Robustness in statistical language modeling: review and perspectives. In: Junqua J, van Noods G (eds) Robustness in language and speech technology. Kluwer, Dordrecht/Boston, pp 101–121

Bellegarda J (2003) Statistical language model adaptation: review and perspectives. Speech Commun 42:93–108

Bisani M, Ney H (2008) Joint sequence models for grapheme-to-phoneme conversion. Speech Commun 50(5):434–451

Blackwood G, de Gispert A, Byrne W (2008) Phrasal segmentation models for statistical machine translation. In: Proceedings of 22nd international conference on computational linguistics (COLING), Manchester

Blain F, Senellart J, Schwenk H, Plitt M, Roturier J (2011) Qualitative analysis of post-editing for high quality machine translation. In: Proceedings of MTS: 13th machine translation summit, Xiamen, pp 164–171

Bontcheva K (2005) Generating tailored textual summaries from ontologies. In: The semantic web: research and applications, Springer, pp 531–545

Brown RD (1996) Example-based machine translation in the Pangloss system. In: Proceedings of the 16th international conference on computational linguistics (COLING-96), Copenhagen, pp 169–174

Brown PF, Della Pietra SA, Della Pietra VJ, Goldsmith MJ, Hajic J, Mercer RL, Mohanty S (1993) But dictionaries are data too. In: Human language technology: proceedings of a workshop held at Plainsboro, New Jersey, USA, Morgan Kaufmann, San Francisco, 21–24 March 1993, pp 202–205

Cai S, Chiang D, Goldberg Y (2011) Language-independent parsing with empty elements. In: Proceedings of the 49th annual meeting of the ACL, Portland, pp 212–216

Callison-Burch C, Koehn P, Osborne M (2006) Improved statistical machine translation using paraphrases. In: HLT-NAACL 2006: proceedings of the human language technology conference of the North American chapter of the ACL, New York, June 2006, pp 17–24

Carl M, Way A (eds) (2003) Recent advances in example-based machine translation. Kluwer, Dordrecht. Introduction to the workshop on EBMT, xxxi

Carpuat M, Diab M (2010) Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In: Proceedings of HLT-NAACL 2010, Los Angeles, pp 242–245

Carpuat M, Wu D (2007) Improving statistical machine translation using word sense disambiguation. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2007), Prague, pp 61–72

Caseli HM, Nunes MGV, Forcada ML (2006) Automatic induction of bilingual resources from aligned parallel corpora: application to shallow-transfer machine translation. Mach Trans 20(4):227–245

Cettolo M, Bertoldi N, Federico M (2011) Methods for smoothing the optimizer instability in SMT. In: Proceedings of the 13th machine translation summit, Asia-Pacific Association for Machine Translation, pp 32–39

Cettolo M, Girardi C, Federico M (2012) WIT3: web inventory of transcribed and translated talks. In: Proceedings of EAMT, Trento, Italy, pp 261–268

Chang Pi-Chuan, Huihsin Tseng, Jurafsky D, Manning CD (2011) Discriminative reordering with Chinese grammatical relations features. In: Proceedings of SSST-3, third workshop on syntax and structure in statistical translation, pp 51–59

Chao Wang, Collins M, Koehn P (2007) Chinese syntactic reordering for statistical machine translation. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Prague, Czech Republic, pp 737–745

Chappelier J-C, Rajman M (1998) A generalized CYK algorithm for parsing stochastic CFG. In: Proceedings of tabulation in parsing and deduction (TAPD'98), Paris, France

Chen Stanley F, Goodman JT (1998) An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University

Chen Yu, Eisele A (2010) Integrating a rule-based with a hierarchical translation system. In: LREC 2010: proceedings of the seventh international conference on language resources and evaluation, Valletta, Malta, 17–23 May 2010, pp 1746–1752

Chenqing Zong, Bo Xu, Taiyi Huang (2002) Interactive Chinese-to-English speech translation based on dialogue management. In: Proceedings of the workshop on speech-to-speech translation: algorithms and systems, pp 61–68

Cherry C, Lin D (2003) A probability model to improve word alignment. ACL-2003: 41st annual meeting of the Association for Computational Linguistics, Sapporo, Japan, 7–12 July 2003

Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: Proceedings of ACL, pp 263–270 (Best paper award)

Chiang D (2007) Hierarchical phrase-based translation. Comput Linguist 33(2):202–228

Chiang D (2010) Learning to translate with source and target syntax. In: Proceedings of ACL10, Stroudsburg, PA, USA, pp 1443–1452

Chiang D, Marton Y, Resnik P (2008) Online large-margin training of syntactic and structural translation features. In: EMNLP 2008: proceedings of the 2008 conference on empirical methods in natural language processing, Honolulu, 25–27 Oct 2008, pp 224–233

Chiang D, Knight K, Wang W (2009) 11,001 new features for statistical machine translation. In: Proceedings of HLT-NAACL09, Boulder, pp 218–226
Choi JD, Palmer M, Nianwen Xue (2009) Using parallel propbanks to enhance word-alignments. In: Proceedings of ACL-IJCNLP workshop on linguistic annotation (LAW'09), pp 121–124
Chung Tagyoung, Gildea D (2010) Effects of empty categories on machine translation. In: Proceedings of EMNLP, pp 636–645
Cicekli I, Altay Güveniri H (2001) Learning translation templates from bilingual translation examples. Appl Intell 15(1):57–76
Cimiano P, Montiel-Ponsoda E, Buitelaar P, Espinoza M, Gomez-Pérez A (2010) A note on ontology localization. J Appl Ontology 5:127–137
Clark J, Dyer C, Lavie A, Smith N (2011) Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In: Proceedings of ACL, Portland
Collier N, Doan S, Kawazoe A, Goodwin RM, Conway M, Tateno Y, Ngo Q-H, Dien D, Kawtrakul A, Takeuchi K, Shigematsu M, Taniguchi K (2008) Bio-Caster: detecting public health rumors with a web-based text mining system. Bioinformatics 24(24):2940–2941
Cowan B, Kučerová I, Collins M (2006) A discriminative model for tree-to-tree translation. In: EMNLP-2006: proceedings of the 2006 conference on empirical methods in natural language processing, Sydney, July 2006, pp 232–241
Cui Lei, Dongdong Zhang, Mu Li, Ming Zhou, Tiejun Zhao (2010) A joint rule selection model for hierarchical phrase-based translation. In: ACL 2010: the 48th annual meeting of the association for computational linguistics, Proceedings of the conference short papers, Uppsala, 11–16 July 2010, pp 6–11
Cui Lei, Dongdong Zhang, Mu Li, Ming Zhou (2011) Function word generation in statistical machine translation systems. MTS:139–146
Daelemans W, Hoste V (eds) (2009) Evaluation of translation technology. Artesis University College, Department of Translators & Interpreters, Antwerp, 261 p
Dan Melamed I (2004) Statistical machine translation by parsing. In: Proceedings of the ACL 2004: 42nd annual meeting of the association for computational linguistics, Barcelona, 21–26 July 2004, pp 653–660
Dandapat S, Forcada ML, Groves D, Penkale S, Tinsley J, Way A (2010) OpenMaTrEx: a free/open-source marker-driven example-based machine translation system. In: Loftsson H et al (eds) Advances in natural language processing: 7th international conference on NLP, IceTAL 2010, Reykjavík, 16–18 Aug 2010. College lecture notes in artificial intelligence, vol 6233. Springer, Berlin/Heidelberg, pp 121–126
Dandapat S, Morrissey S, Way A, Forcada ML (2011) Using example-based MT to support statistical MT when translating homogeneous data in a resource-poor setting, 2011. In: Forcada ML, Depraetere HS, Vandeghinste V (eds) Proceedings of the 15th conference of the European association for machine translation, pp 201–208
DARPA project commented at: 1. http://www.slate.com/articles/technology/future_tense/2012/05/darpa_s_transtac_bolt_and_other_machine_translation_programs_search_for_meaning_.html. 2. http://www.darpa.mil/Our_Work/I2O/Personnel/Dr_Bonnie_Dorr.aspx. 3. http://www.darpa.mil/NewsEvents/Releases/2011/2011/04/19_DARPA_initiates_overarching_language_translation_research_Publishes_Broad_Agency_Announcement_for_Broad_Operational_Language_Translation_program.aspx
David C (2007) Hierarchical phrase-based translation. Comput Linguist 33(2):202–228
David C (2010) Learning to translate with source and target syntax. In: Proceedings of ACL10, Morristown, pp 1443–1452
de Marneffe M-C, Manning CD (2008) Stanford typed hierarchies representation. In: Proceedings of the COLING workshop on cross-framework and cross-domain parser evaluation
de Marneffe, M-C, MacCartney B, Manning CD (2006) Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC-06
Declerck T, Krieger H-U, Thomas SM, Buitelaar P, O'Riain S, Wunner T, Maguet G, McCrae J, Spohr D, Montiel-Ponsoda E (2010) Ontology-based multilingual access to financial reports for sharing business knowledge across Europe. In: Rooz J, Ivanyos J (eds) Internal financial

control assessment applying multilingual ontology framework, HVG Press Kft., Budapest, pp 67–76
Delmonte R (2009b) A computational approach to implicit entities and events in text and discourse. In: International journal of speech technology (IJST), Springer, pp 1–14
Delmonte R (2009a) Understanding implicit entities and events with Getaruns. In: ICSC, 2009 IEEE international conference on semantic computing, Berkeley, pp 25–32
Delmonte R, Bristot A, Tonelli S (2007) VIT – Venice Italian Treebank: syntactic and quantitative features. In: De Smedt K, Hajic J, Kübler S (eds) Proceedings of sixth international workshop on Treebanks and linguistic theories, Nealt proceedings series vol 1, ISSN 1736–6305, pp 43–54
DeNeefe S, Knight K, Wang W, Marcu D (2007) What can syntax-based MT learn from phrase-based MT? In: Proceedings of EMNLP-CoNLL, pp 755–763
Denkowski M, Lavie A (2010) METEOR-NEXT and the METEOR paraphrase tables: improved evaluation support for five target languages. In: ACL 2010: joint fifth workshop on statistical machine translation and MetricsMATR. Proceedings of the workshop, Uppsala University, Uppsala, 15–16 July 2010, pp 339–342
Deyi Xiong, Min Zhang, Haizhou Li (2010) Learning translation boundaries for phrase-based decoding. In: Proceedings of HLT-NAACL 2010
Dienes P, Dubey A (2003) Antecedent recovery: experiments with a trace tagger. In: Proceedings of EMNLP, Sapporo
Ding Yuan, Palmer M (2005) Machine translation using probabilistic synchronous dependency insertion grammars. In: ACL-2005: 43rd annual meeting of the association for computational linguistics, University of Michigan, Ann Arbor, 25–30 June 2005, pp 541–548
Doddington G (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Marcus M (ed) HLT 2002: human language technology conference: proceedings of the second international conference on human language technology research, San Diego, 24–27 Mar 2002, [Morgan Kaufmann for DARPA, San Francisco], pp 138–145
Dugast L, Senellart J, Koehn P (2007) Statistical post-editing on SYSTRAN's rule-based translation system. In: Proceedings of the second workshop on statistical machine translation, Prague, pp 220–223
Dugast L, Senellart J, Koehn P (2008) Can we relearn an RBMT system? In: Proceedings of the third workshop on statistical machine translation, Columbus, pp 175–178
Dugast L, Senellart J, Koehn P (2009) Selective addition of corpus-extracted phrasal lexical rules to a rule-based machine translation system. In: MT Summit XII: proceedings of the twelfth machine translation summit, Ottawa, 26–30 Aug 2009, pp 222–229
Dyer C, Muresan S, Resnik P (2008) Generalizing word lattice translation. In: Proceedings of ACL, Columbus, pp 1012–1020
Ebling S, Way A, Volk M, Naskar SK (2011) Combining semantic and syntactic generalization in example-based machine translation. In: Forcada ML, Depraetere H, Vandeghinste V (eds) Proceedings of the 15th conference of the European association for machine translation, Leuven, pp 209–216
Eisele A, Federmann C, Uszkoreit H, Saint-Amand H, Kay M, Jellinghaus M, Hunsicker S, Herrmann T, Chen Y (2008) Hybrid machine translation architectures within and beyond the Euro Matrix project. In: Hutchins J, von Hahn W (eds) Proceedings of EAMT 2008: 12th annual conference of the European association for machine translation, Hamburg, 22–23 Sept 2008, pp 27–34
Eisner J (2003) Learning non-isomorphic tree mappings for machine translation. ACL-2003: 41st annual meeting of the association for computational linguistics, Sapporo, 7–12 July 2003
El Kholy A, Habash N (2010) Orthographic and morphological processing for English-Arabic statistical machine translation. In: Proceedings of Traitement Automatique du Langage Naturel (TALN-10). Montréal, Canada
España-Bonet C, Jesús G, Lluís M (2009) Discriminative phrase-based models for Arabic machine translation. ACM Trans Asian Lang Info Process J 8(4):1–20
España-Bonet C, Labaka G, Dìaz de Ilarraza A, Màrquez L (2011) Hybrid machine translation guided by a rule-based system, MTS, pp 554–561

Espinoza M, Gomez-Pérez A, Mena E (2008) Enriching an ontology with multilingual information. In: Proceedings of the 5th annual of the European semantic web conference (ESWC08), Tenerife, pp 333–347

Espinoza M, Montiel-Ponsoda E, Gomez-Pérez A (2009). Ontology localization. In: Proceedings of the 5th international conference on knowledge capture (KCAP09), pp 33–40

Esplà-Gomis M, Saànchez-Cartagena VM, Pérez-Ortiz JA (2011) Multimodal building of monolingual dictionaries for machine translation by non-expert users, In: Proceedings of the 13th MTS, Xiamen, pp 147–154

Fai Wong, Dong-Cheng Hu, Yu-Hang Mao, Ming-Chui Dong, Yi-Ping Li (2005) Machine translation based on constraint-based synchronous grammar. In: Proceedings of the 2nd international joint conference on natural language, Jeju Island, Republic of Korea, pp 612–623

Federmann C, Eisele A, Uszkoreit H, Chen Y, Hunsicker S, Xu J (2010) Further experiments with shallow hybrid MT systems. In: ACL 2010: joint fifth workshop on statistical machine translation and MetricsMATR. Proceedings of the workshop, Uppsala University, Uppsala, 15–16 July 2010, pp 77–81

Feifei Zhai, Jiajun Zhang, Yu Zhou, Chengqing Zong (2011) Simple but effective approaches to improving tree-to-tree model, MTS

Font-Llitjòs A, Carbonell JG, Lavie A (2005) A framework for interactive and automatic refinement of transfer-based machine translation. In: European association of machine translation (EAMT) 10th annual conference, Budapest, Hungary, Citeseer

Forcada ML, Depraetere H, Vandeghinste V (eds) (2011) Proceedings of the 15th conference of the European association for machine translation, Leuven, pp 13–20

Forcada ML, Ginest ı-Rosell M, Nordfalk J, O'Regan J, Ortiz-Rojas S, Pérez-Ortiz JA, Sànchez-Martìnez F, Ramìrez-Sànchez G, Tyers FM (2011a) Apertium: a free/open-source platform for rule-based machine translation. Machine translation. Special issue on free/open-source machine translation (in press)

Forcada ML, Ginest **á**-Rosell M, Nordfalk J, O'Regan J, Ortiz-Rojas S, Pérez-Ortiz JA, Sànchez-Martìnez F, Ramìrez-Sànchez G, Tyers FM (2011b) Apertium: a free/open-source platform for rule-based machine translation. Mach Translat (Special Issue on freeo/open-source machine translation) 25(2):127–144.

Fraser A, Marcu D (2006) Semi-supervised training for statistical word alignment. In: Proceedings of ACL, Sydney, pp 769–776

Fraser A, Marcu D (2007) Getting the structure right for word alignment: LEAF. In: Proceedings of EMNLP, Prague, pp 51–60

Fraser A, Marcu D (2007b) Measuring word alignment quality for Statistical Machine Translation. Comput Linguist 33(3):293–303

Fu B, Brennan R, O'Sullivan D (2010) Cross-lingual ontology mapping and its use on the multilingual semantic web. In: Proceedings of the 1st workshop on the multilingual semantic web, at the 19th international World Wide Web Conference (WWW 2010)

Gales MJF, Liu X, Sinha R, Woodland PC, Yu K, Matsoukas S, Ng T, Nguyen K, Nguyen L, Gauvain J-L, Lamel L, Messaoudi A (2007) Speech recognition system combination for machine rranslation. In: IEEE international conference on acoustics, speech and signal processing, Honolulu, pp 1277–1280

Galley M, Hopkins M, Knight K, Marcu D (2004) What's in a translation rule? In: HLT-NAACL 2004: human language technology conference and North American chapter of the association for computational linguistics annual meeting, The Park Plaza Hotel, Boston, 2–7 May 2004, pp 273–280

Galley M, Graehl J, Knight K, Marcu D, DeNeefe S, Wang Wei, Thayer I (2006) Scalable inference and training of context-rich syntactic translation models. In: Proceedings of the international conference on computational linguistics/association for computational linguistics (COLING/ACL-06), Sydney, pp 961–968

Gao Q, Vogel S (2011) Utilizing target-side semantic role labels to assist hierarchical phrase-based machine translation. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, ACL HLT 2011, Portland, June 2011, pp 107–115

Gao Yuqing, Bowen Zhou, Weizhong Zhu, Wei Zhang (2008) Handheld speech to speech translation system. Automatic speech recognition on mobile devices and over communication networks, Springer, London

Gildea D (2003) Loosely tree-based alignment for machine translation ACL-2003.In: 41st annual meeting of the association for computational linguistics, Sapporo, 7–12 July 2003

Gough N, Way A (2003) Controlled generation in example-based machine translation MT Summit IX, New Orleans, 23–27 Sept 2003, pp 133–140

Green T (1979) The necessity of syntax markers: two experiments with artificial languages. J Verbal Learn Behav 18:481–496

Habash N, Dorr B, Monz C (2009) Symbolic-to-statistical hybridization: extending generation-heavy machine translation. Mach Transl 23:23–63

Hanneman G, Lavie A (2009) Decoding with syntactic and non-syntactic phrases in a syntax-based machine translation system, 2009. In: Proceedings of SSST-3, third workshop on syntax and structure in statistical translation, ACL, pp 1–9

Haque R, Naskar SK, Ma Y, Way A (2009) Using supertags as source language context in SMT. In: Lluís Màrquez, Harold Somers (eds) EAMT-2009: proceedings of the 13th annual conference of the European association for machine translation, Universitat Politècnica de Catalunya, Barcelona, 14–15 May 2009, pp 234–241

He Xiaodong, Mei Yang, Jianfeng Gao, Patrick Nguyen, Robert Moore (2008) Indirect HMM based hypothesis alignment for combining outputs from machine translation systems. In:Proceedings of the 2008 conference on empirical methods in natural language processing, pp 98–107

Hermjakob Ulf (2009) Improved Word alignment with statistics and linguistic heuristics. In: Proceedings of EMNLP 2009, Singapore, pp 229–237

Heyn M (1996) Integrating machine translation into translation memory systems. In: Proceedings of the EAMT machine translation workshop, TKE '96, Vienna, pp 113–126

Hoang H, Koehn P (2010) Improved translation with source syntax labels. In: Proceedings of the joint 5th workshop on statistical machine translation and metrics MATR, Uppsala, 11–16 July 2010, pp 409–417

Hoang H, Koehn P, Lopez A (2009) A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In: IWSLT 2009: proceedings of the international workshop on spoken language translation, National Museum of Emerging Science and Innovation, Tokyo, 1–2 Dec 2009, pp 152–159

Hovy E (1998) Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In: Proceedings of the first international conference on language resources and evaluation, Granada

Hovy E, Nirenburg S (1992) Approximating and interlingua in a principled way. In: Proceedings of the DARPA speech and natural language workshop, Arden House

Hovy E, Marcus M, Palmer M, Pradhan S, Ramshaw I, Weischedel R (2006) OntoNotes: the 90 % solution. In: Proceedings of the human language technology conference at the annual meeting of NAACL, New York

Huang Liang, David Chiang (2007) Forest rescoring: faster decoding with integrated language models. In: ACL 2007: proceedings of the 45th annual meeting of the association for computational linguistics, Prague, June 2007, pp 144–151

Huang Liang, Knight K, Joshi A (2006) Statistical syntax-directed translation with extended domain of locality. In: AMTA 2006: proceedings of the 7th conference of the association for machine translation in the americas, visions for the future of machine translation, Cambridge, MA, 8–12 Aug 2006, pp 66–73

Huang Liang, Hao Zhang, Daniel Gildea, Kevin Knight (2009) Binarization of synchronous context-free grammars. Comput Linguist 35(4):559–595

Huang Chu-Ren, Ru-Yng Chang, Hsiang-bin Lee (2010) Sinica BOW (Bilingual Ontological WordNet): integration of bilingual WordNet and SUMO. In: Huang et al (eds) Ontology and the lexicon, CUP, Cambridge, pp 201–211

Huang Zhongqiang, Martin Cmejrek, Bowen Zhou (2010) Soft syntactic constraint for hierarchical phrase-based translation using latent syntactic distributions. In: Proceedings of EMNLP10, Cambridge, MA

Hutchins J (2005a) State of the art reports natural language translation computer-based translation systems and tools. www.hutchingsweb.me.uk/BCS-NLT-2005.pdf

Hutchins J (2005b) Current commercial machine translation systems and computer-based translation tools: system types and their uses. Int J Transl 17(1–2):5–38

Hutchins J (2010) Outline of machine translation developments in Europe and America. JAPIO, Tokyo, pp 1–8

Hutchins WJ, Somers HL (1992) An introduction to machine translation. Academic, London, Xxi, 362pp

Hwidong Na, Jong-Hyeok Lee (2011) Multi-word unit dependency forest-based translation rule extraction. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, ACL HLT 2011, pp 41–51

Hyoung-Gyu Lee, Min-Jeong Kim, Gumwon Hong, Sang-Bum Kim, Young-Sook Hwang, Hae-Chang Rim (2010) Identifying idiomatic expressions using phrase alignments in bilingual parallel corpus. In: Proceedings of PRICAI 2010, Daegu, Korea

Jelinek F (2004) Stochastic analysis of structured language modeling. In: Johnson M, Khudanpur S, Ostendorf M, Rosenfeld R (eds) Mathematical foundations of speech and language processing. Springer, Berlin, pp 37–72

Jianjun Ma, Degen Huang, Haixia Liu, Wenfeng Sheng (2011) POS tagging of English particles for machine translation. In: Proceedings of the 13th machine translation summit, Asia-Pacific Association for Machine Translation, Xiamen, pp 57–63

Jijkoun V, de Rijke M (2004) Enriching the output of a parser using memory-based learning. In: Proceedings of the 42nd annual meeting of the Association for Computational Linguistics

Johannessen JB, Nordgård T, Nygaard L (2008) Evaluation of linguistics-based translation. In: LREC 2008: 6th language resources and evaluation conference, Marrakech, pp 26–30

Johansson R, Nugues P (2007) Extended constituent-to-dependency conversion for english. In: Proceedings of NODALIDA 2007, Tartu

Josef OF, Ney H (2003) A systematic comparison of various statistical alignment models. Comput Linguist 29(1):19–51

Josef OF, Ney H (2004) The alignment template approach to statistical machine translation. Comput Linguist 30(4):417–449

Karakos D, Khudanpur S (2008) Sequential system combination for machine translation of speech. In: Proceedings of the 2008 IEEE workshop on spoken language technology (SLT-08), Goa

Kay M (2011) Zipf's Law and *L'Arbitraire du Signe*. Linguist Issues Lang Technol (LiLT) 6(8):1–25, CLSI Publications

Khadivi S, Zens R, Ney H (2006) Integration of speech to computer-assisted translation using finite-state automata. In: Proceedings of COLING/ACL 2006, Sydney

Khalilov M, Pretkalniņa L, Kuvaldina N, Pereseina V (2010) SMT of Latvian, Lithuanian and Estonian languages: a comparative study. In: Human language technologies – the Baltic perpective, international conference, Riga, 8 Oct 2010, 8pp

Kirchhoff K, Rambow O, Habash N, Diab M (2007) Semi-automatic error analysis for large-scale statistical machine translation systems. In: Proceedings of the machine translation summit (MT- Summit), Copenhagen

Klein D, Manning CD (2003) Accurate unlexicalized parsing. In: ACL 41, pp 423–430

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP, Barcelona, pp 388–395

Koehn P (2010) Statistical machine translation. Cambridge University Press, Cambridge, XII, 433p

Koehn P, Knight K (2003) Feature-rich statistical translation of noun phrases. In: Proceedings of ACL 2003, Hongkong

Koehn P, Monz C (2006) Manual and automatic evaluation of machine translation between European languages. In: NAACL 2006 workshop on statistical machine translation, ACL, New York, pp 102–121

Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: Proceedings of NAACL, Morristown, pp 48–54

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pp 177–180

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Labaka G (2010) EUSMT: incorporating linguistic information into SMT for a morphologically rich language. Its use in SMT-RBMT-EBMT hybridation. Ph.D. thesis, University of the Basque Country

Labaka G (2010) EUSMT: incorporating linguistic information into SMT for a morphologically rich language. Its use in SMT-RBMT-EBMT hybridation. Ph.D. thesis, University of the Basque Country

Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML, pp 282–289

Lagarda A-L, Alabau V, Casacuberta F, Silva R, Díaz-de-Liaño E (2009) Statistical post-editing of a rule-based machine translation system. In: Proceedings of NAACL HLT 2009. Human language technologies: the 2009 annual conference of the North American chapter of the ACL, short papers, Boulder, 31 May–5 June 2009, pp 217–220

Lambert P, Banchs R (2005) Data inferred multi-word expressions for statistical machine translation. In: Proceedings of MT summit X, Phuket

Lamel L et al (2011) Speech recognition for machine translation in quaero. In: Proceedings of IWSLT, San Francisco

Lavie A (2008) Stat-XFER: a general search-based syntax-driven framework for machine translation. In: Computational linguistics and intelligent text processing, Springer, LNCS, New York, pp 362–375

Lavie A, Agarwal A (2007) METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In: Proceedings of the second workshop on statistical machine translation, Prague, pp 228–231

Lavie A, Waibel A et al (1996) Translation of conversational speech with JANUS-II. In: Proceedings of ICSLP 96, Philadelphia

Lavie A et al (2002) A multi-perspective evaluation of the NESPOLE! speech-to-speech translation system. In: Proceedings of ACL 2002 workshop on speech-tospeech translation: algorithms and systems, Philadelphia

Lavie A, Yarowsky D, Knight K, Callison-Burch C, Habash N, Mitamura T (2006) MINDS workshops machine translation working group final report. http://www-nlpir.nist.gov/MINDS/FINAL/MT.web.pdf

Lavie A, Parlikar A, Ambati V (2008) Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In: Proceedings of the second ACL workshop on syntax and structure in statistical translation, Columbus, pp 87–95

Lei Cui, Zhang D, Li M, Zhou M, Zhao T (2010) A joint rule selection model for hierarchical phrase-based translation. In: Proceedings of ACL10, Uppsala, pp 6–11

Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou (2011) Function word generation in Statistical Machine Translation Systems, MTS 2011, pp 139–146

Lemao Liu, Tiejun Zhao, Chao Wang, Hailong Cao (2011) A unified and discriminative soft syntactic constraint model for hierarchical phrase-based translation

Levin L, Lavie A, Woszczyna M, Gates D, Galvadá M, Koll D, Waibel A (2000) The janus-III translation system: speech-to-speech translation in multiple domains. Mach Trans 15:3–25

Levy R, Manning C (2004) Deep dependencies from context–free statistical parsers: correcting the surface dependency approximation. In: Proceedings of the ACL

Li Zhifei, Callison-Burch C, Dyer C, Ganitkevitch J, Khudanpur S, Schwartz L, Thornton WNG, Weese J, Zaidan OF (2009) Demonstration of Joshua: an open source toolkit for parsing-based machine translation. In: Proceedings of the 47th annual meeting of the ACL and the 4th IJCNLP, Software demonstrations, Suntec, 3 Aug 2009, pp 25–28

Liu Ding, Gildea D (2008) Improved tree-to-string transducer for machine translation. In: Proceedings of ACL-08: HLT. Third workshop on statistical machine translation (ACL WMT-08), The Ohio State University, Columbus, 19 June 2008, pp 62–69

Liu Ding, Gildea D (2010) Semantic role features for machine translation. In: Proccedings of the coling 2010: 23rd international conference on computational linguistics, Beijing International Convention Center, Beijing, 23–27 Aug 2010, pp 716–724

Liu Yang, Qun Liu, Shouxun Lin (2006) Tree-to-string alignment template for statistical machine translation. In: Coling-ACL 2006: proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics, Sydney, 17–21 July 2006, pp 609–616

Liu Yang, Yajuan Lü, Qun Liu (2009) Improving tree-to-tree translation with packed forests. In: Proceedings of the 47th annual meeting of the ACL and the 4th IJCNLP, Suntec, 2–7 Aug 2009, pp 558–566

Liu Zhanyi, Haifeng Wang, Hua Wu, Sheng Li (2010) Improving statistical machine translation with monolingual collocation. In: Proceedings of ACL 2010, Uppsala

Liu Lemao, Tiejun Zhao, Chao Wang, Hailong Cao (2011a) A unified and discriminative soft syntactic constraint model for hierarchical phrase-based translation

Liu Shujie, Chi-Ho Li, Ming Zhou (2011b) A unified SMT framework combining MIRA and MERT in MTS, pp 181–188

Liu Yang, Qun Liu, Yajuan Lü (2011) Adjoining tree-to-string translation. In: ACL-HLT 2011: proceedings of the 49th annual meeting of the association for computational linguistics, Portland, 19–24 June 2011, pp 1278–1287

Llitjós AF, Vogel S (2007) A walk on the other side. Adding statistical components to a transfer-based translation system. In: Proceedings of the HLT-NAACL workshop on syntax and structure in statistical translation, Rochester, pp 72–79

Lopez A, Resnik P (2006) Word-based alignment, phrase-based translation: what's the link. In: Proceedings of AMTA, pp 90–99

Lopez A, Resnik P (2006) Word-based alignment, phrase-based translation: what's the link. In: Proceedings of the AMTA, Cambridge MA, pp 90–99

Maletti A (2010) Why synchronous tree substitution grammars? In: Proceedings of the 2010 meeting of the North American chapter of the association for computational linguistics (NAACL-10), pp 876–884

Maletti A, Graehl J, Hopkins M, Knight K (2009) The power of extended top-down tree transducers. SIAM J Comput 39:410–430

Marcu D, Wei Wang, Echihabi A, Knight K (2006) SPMT: statistical machine translation with syntactified target language phrases. In: Proceedings of EMNLP 2006, pp 44–52

Mareček D (2009a) Improving word alignment using alignment of deep structures. In: Proceedings of the 12th international conference on text, speech and dialogue, pp 56–63

Mareček D (2009b) Using tectogrammatical alignment in phrase-based machine translation. In: Proceedings of WDS 2009 contributed papers, pp 22–27

Marton Y, Resnik P (2008) Soft syntactic constraints for hierarchical phrased-based translation. Proceedings of the ACL-08: HLT 46th annual meeting of the association for computational linguistics: human language technologies, The Ohio State University, Columbus, 15–20 June 2008, pp 1003–1011

Maruyama H, Watanabe H (1992) Tree cover search algorithm for example-based translation fourth international conference on theoretical and methodological issues in machine translation (TMI-92), empiricist vs. rationalist methods in MT, Montreal, CCRIT-CWARC, 25–27 June 1992, pp 173–184

Matthew GS, Madnani N, Dorr B, Schwartz R (2009) TER-Plus: paraphrase, semantic, and alignment enhancements to translation error rate. Mach Trans 23(2/3):117–127

McCrae J, Campana J, Cimiano P (2010) CLOVA: an architecture for cross-language semantic data querying. In: Proceedings of the first mutlilingual semantic web workshop

McCrae et al (2011) Combining statistical and semantic approaches to the translation of ontologies and taxonomies. In: Proceedings of SSST, pp 116–125

McCrae J, Espinoza M, Montiel-Ponsoda E, Aguado-de-Cea G, Cimiano P (2011) Combining statistical and semantic approaches to the translation of ontologies and taxonomies. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, ACL HLT 2011, Portland, pp 116–125

Menezes A, Quirk C (2008) Syntactic models for structural word insertion and deletion. In: Proceedings of EMNLP

Menezes A, Richardson SD (2001) A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In: Proceedings of MT summit VIII workshop on example-based machine translation, Santiago de Compostela, 18–22 Sept 2001

Menezes A, Toutanova K, Quirk C (2006) Microsoft research treelet translation system: NAACL 2006 Europarl evaluation. In: HLT-NAACL 2006: proceedings of the workshop on statistical machine translation, New York, June 2006, pp 158–161

Mi Haitao, Liang Huang, Qun Liu (2008) Forest-based translation. In: Proceedings of the ACL-08: HLT: 46th annual meeting of the association for computational linguistics: human language technologies, 15–20 June 2008, The Ohio State University, Columbus, pp 192–199

Michael C, Way A (eds) (2003) Recent advances in example-based machine translation. Kluwer Academic, Dordrecht. Introduction to the workshop on EBMT, xxxi

Ming Tan, Wenli Zhou, Lei Zheng, Shaojun Wang (2012) A scalable distributed syntactic, semantic and lexical language model, to appear in computational linguistics just accepted MS, pp 1–66

Mischo W (1982) Library of congress subject headings. Catalog Classif Quart 1(2):105–124

Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pp 177–180

Morimoto T et al ATR's speech translation system: ASURA. In: Proceedings of EuroSpeech 93, pp 1291–1294

Na Hwidong, Lee Jong-Hyeok (2011) Multi-word unit dependency forest-based translation rule extraction. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, ACL HLT 2011, pp 41–51

Nakazawa T, Kurohashi S (2008) Linguistically-motivated tree-based probabilistic phrase alignment. In: AMTA-2008 MT at work: proceedings of the eighth conference of the association for machine translation in the Americas, Waikiki, 21–25 Oct 2008, pp 163–171

Nakazawa T, Kurohashi S (2011) Statistical phrase alignment model using dependency relation probability. In: Proceedings of SSST-3, third workshop on syntax and structure in statistical translation, pp 10–18

Navigli R, Ponzetto SP (2010) Babelnet: building a very large multilingual semantic network. In: Proceedings of the 48th annual meeting of the association for computational linguistics, pp 216–225

Nießen S, Och FJ, Leusch G, Ney H (2000) An evaluation tool for machine translation: fast evaluation for MT research. In: Proceedings of LREC-2000: second international conference on language resources and evaluation, Athens, 31 May–2 June 2000, pp 39–45

Och FJ (2003) Minimum error rate training in statistical machine translation. In: Proceedings of ACL, pp 160–167

Och FJ, Ney H (2002) Discriminative training and maximum entropy models for statistical machine translation. In: ACL 2002: proceedings of the 40th annual meeting of the association for computational linguistics (best paper award), Philadelphia, July 2002, pp 295–302

Och FJ, Ney H (2003a) A systematic comparison of various statistical alignment models. Comput Linguist 29(1):19–51

Och FJ, Ney H (2003b) The alignment template approach to statistical machine translation. Comput Linguist 30(4):417–449

Oepen S, Lønning JT (2006) Discriminant-based MRS banking. In: Proceedings of the 4th international conference on language resources and evaluation

Oepen S, Dyvik H, Lønning JT, Velldal E, Beermann D, Carroll J, Flickinger D, Hellan L, Johannessen JB, Meurer P, Nordgård T, Rosén V (2004) Som a kapp-ete med trollet? Towards MRS-based

Norwegian–english machine translation. In: Proceedings of the 10th international conference on theoretical and methodological issues in machine translation, Baltimore, pp 11–20
Oepen S, Dyvik H, Flickinger D, Lønning JT, Meurer P, Rosén V (2005) Holistic regression testing for high-quality MT. Some methodological and technological reflections. In: Proceedings of the 10th annual conference of the European association for machine translation, Budapest
Oepen S, Velldal E, Lønning JT, Meurer P, Rosén V, Flickinger D (2007) Towards hybrid quality-oriented machine translation—on linguistics and probabilities in MT. In: TMI-2007: proceedings of the 11th international conference on theoretical and methodological issues in machine translation, Skövde, 7–9 Sept 2007, pp 144–153
Oflazer K, El-Kahlout ID (2007) Exploring different representational units in English-to-Turkish statistical machine translation. In: Proceedings of the second workshop on statistical machine translation, ACL, pp 25–32
Owczarzak K, van Genabith J (2007) Evaluating machine translation with LFG dependencies [abstract]. Mach Trans 21(2):95–119
Papineni K, Roukos S, Ward T, Wei-jing Zhu (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of ACL
Pekar V, Mitkov R, Blagoev D, Mulloni A (2006) Finding translations for low-frequency words in comparable corpora. Mach Transl 20(4):247–266
Petrov S, Barrett L, Thibaux R, Klein D (2006) Learning accurate, compact, and interpretable tree annotation. In: Proceedings of COLING-ACL
Phillips AB (2011) Cunei: open-source machine translation with relevance-based models of each translation instance, in special issue: free/open-source machine translation machine translation. Mach Trans 25(2):161–177
Philpot A, Hovy E, Pantel P (2010) The OMEGA ontology. In: Huang CR et al (eds) Ontology and the lexicon. Cambridge University Press, Cambridge, UK, pp 258–270
Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, Manning CD (2011) Discriminative reordering with Chinese grammatical relations features. In: Proceedings of SSST-3, third workshop on syntax and structure in statistical translation, Boulder, pp 51–59
Pighin Daniele, Lluìs Màrquez (2011) Automatic projection of semantic structures: an application to pairwise translation Popović, Maja & Hermann Ney: 2006. POS-based reorderings for statistical machine translation. In: LREC-2006: fifth international conference on language resources and evaluation, Genoa, 22–28 May 2006, pp1278–1283
Pighin D, Màrquez L (2011) Automatic projection of semantic structures: an application to pairwise translation ranking. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical rranslation, Portland, pp 1–9
Przybocki M, Peterson K, Bronsart S (2008) Translation adequacy and preference evaluation tool (TAP-ET). In: LREC 2008: 6th language resources and evaluation, Marrakech, 26–30 May 2008, 8pp
Qin Gao, Vogel S (2011) Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, ACL HLT 2011, Portland, pp 107–115
Quirk C, Menezes A, Cherry C (2005) Dependency treelet translation: syntactically informed phrasal SMT. In: ACL-2005: 43rd annual meeting of the association for computational linguistics, University of Michigan, Ann Arbor, 25–30 June 2005, pp 271–279
Ranking, in proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, pp 1–9
Ravi S, Knight K (2010) Does GIZA++ make search errors? Comput Linguist Squibs Discuss 36(3):295–302
Ren, Zhixiang, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang (2009) Improving statistical machine translation using domain bilingual multiword expressions. In: Proceedings of MWE 2009 (ACL-IJCNLP)
Riezler S, Maxwell JT III (2006) Grammatical machine translation. In: Proceedings of the human language technology conference and annual meeting of the North American association for computational linguistics, pp 248–255

Rosti AVI, Bing Xiang, Matsoukas S, Schwartz R, Ayan NF, Dorr BJ (2007a) Combining outputs from multiple machine translation systems. In: Proceedings of HLT-NAACL, Rochester, pp 228–235

Rosti AVI, Matsoukas S, Schwartz R (2007b) Improved word-level system combination for machine translation. In: Proceedings of ACL-07, pp 312–319

Rosti AVI, Bing Zhang, Matsoukas S, Schwartz R (2008) Incremental hypothesis alignment for building confusion networks with application to machine translation system combination. In: Proceedings of ACL/WMT 2008, pp 183–186

Roturier J (2009) Deploying novel MT technology to raise the bar for quality: a review of key advantages and challenges. Keynote slides, the twelfth machine translation summit, International association for machine translation, Ottawa

Saers M (2011) Translation as linear transduction: models and algorithms for efficient learning in statistical machine translation. Ph.D. thesis, Uppsala University, Department of Linguistics and Philology, Uppsala

Saers M, Nivre J, Wu D (2010) Word alignment with stochastic bracketing linear inversion transduction grammar. In: HLT/NAACL2010, ACL, pp 341–344

Sag IA, Baldwin T, Bond F, Copestake A, Flickinger D (2001) Multiword expressions: a pain in the neck for NLP. In: Proceedings of WP

Sanchez-Cartagena VM, Pérez-Ortiz JA (2010) Tradubi: open-source social translation for the Apertium machine translation platform. In: Open source tools for machine translation, MT Marathon 2010, pp 47–56

Sanchez-Cartagena VM, Sanchez-Martìnez F, Pérez-Ortiz JA (2011) Integrating shallow-transfer rules into phrase-based statistical machine translation, MT Summit XIII: the thirteenth machine translation summit [organized by the] Asia-Pacific association for machine translation (AAMT), Xiamen, 19–23 Sept 2011, pp 562–569

Sanchez-Martinez F, Forcada ML (2009) Inferring shallow-transfer machine translation rules from small parallel corpora. J Artif Intell Res 34:605–635

Schafer C, David Y (2003) Statistical machine translation using coercive two-level syntactic transduction EMNLP-2003. In: proceedings of the 2003 conference on empirical methods in natural language processing, a meeting of SIGDAT, a special interest group of the ACL, held in conjunction with ACL-03, Sapporo, 11–12 July 2003, 8pp

Schafer C, Yarowsky D (2002) Inducing translation lexicons via diverse similarity measures and bridge languages. In: CoNLL, Taipei

Schmid H (2006) Trace prediction and recovery with unlexicalized PCFGs and slash features. In Proceedings of the COLING-ACL, Sydney

Schwenk H, Abdul-Rauf S, Barrault L, Senellart J (2009) SMT and SPE machine translation system for WMT'09. In: Proceedings of the fourth workshop on statistical machine translation, Athens, 30–31 March 2009, pp 130–134

Shen Libin, Jinxi Xu, Weischedel R (2008) A new string-to-dependency machine translation algorithm with a target dependency language model. ACL-08: HLT. In: 46th annual meeting of the association for computational linguistics: human language technologies. Proceedings of the conference, The Ohio State University, Columbus, 15–20 June 2008, pp 577–585

Shen Libin, Jinxi Xu, Bing Zhang, Matsoukas S, Weischedel R (2009) Effective use of linguistic and contextual information for statistical machine translation. EMNLP-2009. In: Proceedings of the 2009 conference on empirical methods in natural language processing, Singapore, 6–7 Aug 2009, pp 72–80

Shu Cai, Chiang D, Goldberg Y (2011) Language-independent parsing with empty elements. In: Proceedings of the 49th annual meeting of the ACL, Portland, pp 212–216

Shujie Liu, Chi-Ho Li, Ming Zhou (2011) A unified SMT framework combining MIRA and MERT in MTS, pp 181–188

Shumin Wu, Palmer M (2011) Semantic mapping using automatic word alignment and semantic role labeling. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, Portland, pp 21–30

Shumin Wu, Choi JD, Palmer M (2010) Detecting cross-lingual semantic similarity using parallel propbanks. In: Proceedings of the 9th conference of the association for machine translation in the Americas

Sim, Khe Chai, William JB, Mark JFG, Hichem S, Phil CW (2007) Consensus network decoding for statistical machine translation system combination. In: Proceedings of the 32nd IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp 105–108

Simard M, Ueffing N, Isabelle P, Kuhn R (2007) Rule-based translation with statistical phrase-based post-editing. In: Proceedings of the second workshop on statistical machine translation, ACL, pp 203–206

Simard M, Cyril G, Pierre I (2007) Statistical phrase-based post-editing. NAACL-HLT-2007 Human language technology: the conference of the North American chapter of the association for computational linguistics, Rochester, 22–27 April 2007, pp 508–515

Smith DA, Eisner J (2006) Minimum risk annealing for training log-linear models. In: Proceedings of the COLING/ACL on main conference poster sessions, ACL, pp 787–794

Snover M, Bonnie D, Richard S, Linnea M, John M (2006) A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th conference of the association for machine translation in the Americas (AMTA-2006), Cambridge, MA, Aug, pp 223–231

Soltau H, Metze F, Fügen C, Waibel A (2001) A one pass-decoder based on polymorphic linguistic context assignment. In: IEEE ASRU, Madonna di Campiglio

Somers H (2003) An overview of EBMT. In: Michael C, Andy W (eds) Recent advances in example-based machine translation. Kluwer Academic, Dordrecht, pp 3–57

Specia L, Cancedda N, Turchi M, Cristianini N (2009) Estimating the sentence-level quality of machine translation systems. In: Proceedings of the 13th annual conference of the EAMT, pp 28–35

Specia L, Saunders C, Turchi M, Wang Z, Shawe-Taylor J (2009) Improving the confidence of machine translation quality estimates. MT summit XII

Stein D, Stephan P, David V, Hermann N (2010) A cocktail of deep syntactic features for hierarchical machine translation. AMTA 2010: the ninth conference of the association for machine translation in the Americas, Denver 31 Oct–4 Nov 2010, 9pp

Stolcke A (2002) SRILM – an extensible language modeling toolkit. In: Proceedings of the international conference of spoken language processing, vol 2, Denver, pp 901–904

Suzuki H (2011) Automatic post-editing based on SMT and its selective application by sentence-level automatic quality evaluation, MTS, pp 156–163

Tan Ming, Wenli Zhou, Lei Zheng, Shaojun Wang (2012) A scalable distributed syntactic, semantic and lexical language model, to appear in computational linguistics Just accepted MS, pp 1–66

Teramusa E (2007) Rule based machine translation combined with statistical post-editor for Japanese to English patent translation. Tokyo University of Science, Suwas

Thurmair G (2009) Comparing different architectures of hybrid machine translation systems. In: Proceedings of MT summit XII

Tiedemann J (2011) Bitext alignment. Morgan & Claypool Publishers, viii, 153 p

Tiedemann J, Kotzé G (2009) Building a large machine-aligned parallel treebank. In: Proceedings of the 8th international workshop on treebanks and linguistic theories (TLT), Milan, pp 197–208

Tillmann C (2004) A unigram orientation model for statistical machine translation. HLT-NAACL 2004: Human language technology conference and North American chapter of the association for computational linguistics annual meeting, The Park Plaza Hotel, Boston, – Short Papers, 2–7 May 2004, pp 101–104

Tinsley J, Hearne M, Way A (2007) Exploiting parallel treebanks to improve phrase-based statistical machine translation. In: Proceedings of the sixth international workshop on treebanks and linguistic theories, pp 175–187

Tonelli S, Delmonte R, Bristot A (2008) Enriching the Venice Italian Treebank with dependency and grammatical relations. In: Proceedings of LREC 2008, Marrakech

Tonelli S, Rodolfo D, Antonella B (2008) Enriching the Venice Italian treebank with dependency and grammatical relations, LREC 2008

Tong Xiao, Jingbo Zhu, Shujie Yao, Hao Zhang (2011) Document-level consistency verification in machine translation, MST 2011, pp 131–138

Turian JP, Luke S, Melamed ID (2003) Evaluation of machine translation and its evaluation MT Summit IX, New Orleans, 23–27 Sept 2003, pp 386–393

Tyers FM (2009) Rule-based augmentation of training data in Breton-French statistical machine translation. In: Proceedings of the 13th annual conference of the European association for machine translation, pp 213–217

Ueffing N, Haffari G, Sarker A (2007) Semi-supervised model adaptation for statistical machine translation. Mach Trans 21(2):77–94

Vandeghinste V, Scott M (2010) Bottom-up transfer in example-based machine translation. In: François I, Veale, T, Andy W (eds) 1997 Gaijin: a bootstrapping, template-driven approach to example-based MT. Proceedings of the 2nd international conference on recent advances in natural language processing, Tzigov Chark1997, pp 239–244

Vandeghinste V, Van den Bogaert J, Martens S, Kotzé G (2011) PaCo-MT: parse and corpus-based machine translation. In: Forcada ML, Depraetere H, Vandeghinste V (eds) Proceedings of the 15th annual conference of the European association for machine translation, p 347

Velldal E, Oepen S (2006) Statistical ranking in tactical generation. In: Proceedings of the conference on empirical methods in natural language processing. Sydney

Velldal E, Oepen S, Flickinger D (2004) Paraphrasing treebanks for stochastic realization ranking. In: Proceedings of the 3rd workshop on treebanks and linguistic theories, pp 149–160

Venkatapathy S, Sangal R, Joshi A, Gali K (2010) A discriminative approach for dependency based statistical machine translation. In: Proceedings of SSST, pp 66–74

Venugopal A, Andreas Z, Noah AS, Stephan V (2009) Preference grammars: softening syntactic constraints to improve statistical machine translation. NAACL HLT 2009. Human language technologies: the 2009 annual conference of the North American chapter of the ACL, Boulder, 31 May–5 June 2009, pp37–45

Viggo H (eds) Proceedings of the 14th international conference of the European association for machine translation (EAMT-2010), 8pp

Vossen P (1998) EuroWordNet: a multilingual database with lexical semantic networks. Kluwer, Dordrecht

Waibel A, Ahmed B, Alan WB, Robert F, Donna Gates AL, Lori L, Kevin L, Laura MT, Juergen R, Tanja S, Dorcas W, Monika W, Jing Z (2003) Speechalator: two-way speech-to-speech translation in your hand. In: Proceedings of HLT-NAACL 2003, Demonstrations, May–June 2003, pp 29–30

Wallach H (2006) Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd international conference on machine learning (ICML), New York, pp 977–984

Wang S, Wang S, Greiner R, Schuurmans D, Cheng L (2005) Exploiting syntactic, semantic and lexical regularities in language modeling via directed Markov random fields. In: Proceedings of the 22nd international conference on machine learning (ICML), Bonn, pp 953–960

Wang S, Wang S, Cheng L, Greiner R, Schuurmans D (2006) Stochastic analysis of lexical and semantic enhanced structural language model. In: Proceedings of the 8th international colloquium on grammatical inference (ICGI), Tokyo, pp 97–111

Wang Chao, Michael Collins, Philipp K (2007) Chinese syntactic reordering for statistical machine translation. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Prague, pp 737–745

Wang Wei, Kevin Knight, Daniel Marcu (2007) Binarizing syntax trees to improve syntax-based machine translation accuracy. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, Prague, pp 746–754

Wei Wang, May J, Knight K, Marcu D (2010) Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. Comput Linguist 36(2):247–277

Watanabe Hideo, Sadao Kurohashi, Eiji Aramaki (2000) Finding structural correspondences from bilingual parsed corpus for corpus-based translation Coling 2000 in Europe: the 18th international conference on computational linguistics. In: Proceedings of the conference, Universität des Saarlandes, Saarbrücken, 31 July–4 Aug 2000, pp 906–912

Wehrli E (2007) Fips, a "deep" linguistic multilingual parser. In: Proceedings of the ACL 2007 workshop on deep linguistic processing, Prague, pp 120–127

Wehrli E, Nerima L, Seretan V, Scherrer Y (2009) On-line and off-line translation aids for non-native readers. In: Proceedings of the international multiconference on computer science and information technology, vol 4, pp 299–303

Wehrli E, Seretan V, Nerima L, Russo L (2009) Collocations in a rule-based MT system: a case study evaluation of their translation adequacy. In: EAMT-2009: proceedings of the 13th annual conference of the European association for machine translation, Lluís Màrquez, Harold Somers (eds), 14–15 May 2009, Universitat Politècnica de Catalunya, Barcelona, pp 128–135

Wei Wang, Knight K, Marcu D (2007) Binarizing syntax trees to improve syntax-based machine translation accuracy. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, pp 746–754

Wilks Y (1994) Stone soup and the French room. In: Zampolli A, Calzolari N, Palmer M (eds) Current issues in computational linguistics: in honour of Don Walker, vol 9–10, Linguistica Computazionale. Giradini Editori/Kluwer Academic, Pisa/Dordrecht, pp 585–594

Wilks Y (2009) Machine translation: its scope and limits. Springer, New York, 252 p

Wong Fai, Dong-Cheng Hu, Yu-Hang Mao, Ming-Chui Dong, Yi-Ping Li (2005) Machine translation based on constraint-based synchronous grammar. In: Proceedings of the 2nd international joint conference on natural language, Jeju Island, pp 612–623

Wu D (1997) Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Comput Ling 23(3):378–403

Wu Dekai, Hongsing Wong (1998) Machine translation with a stochastic grammatical channel. In: Coling-ACL '98: 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics, Université de Montréal, Montreal, 10–14 Aug 1998, pp 1408–1415

Wu Shumin, Martha Palmer (2011) Semantic mapping using automatic word alignment and semantic role labeling. In: Proceedings of SSST-5, fifth workshop on syntax, semantics and structure in statistical translation, Portland, pp 21–30

Wu Dekai, Pascale Fung (2009) Semantic roles for SMT: a hybrid two-pass model. In: NAACL HLT 2009: human language technologies: the 2009 annual conference of the North American chapter of the ACL, Short papers, Boulder, 31 May–5 June 2009, pp 13–16

Wu Shumin, Jinho D Choi, Martha Palmer (2010) Detecting cross-lingual semantic similarity using parallel propbanks. In: Proceedings of the 9th conference of the association for machine translation in the Americas

Wuebker J, Mauser A, Ney H (2010) Training phrase translation models with leaving-one-out. In: Proceeding of ACL, pp 475–484

Tong Xiao, Jingbo Zhu, Shujie Yao, Hao Zhang (2011) Document-level consistency verification in machine translation. MST 2011:131–138

Xiong Deyi, Qun Liu, Shouxun Lin (2006) Maximum entropy based phrase reordering model for statistical machine translation. In: Coling-ACL 2006: proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics, Sydney, 17–21 July 2006, pp 521–528

Xiong D, Zhang M, Li H (2010) Error detection for statistical machine translation using linguistic features. In: ACL 2010: the 48th annual meeting of the association for computational linguistics, Uppsala, pp 604–611

Xiong Deyi, Min Zhang, Haizhou Li (2010a) Learning translation boundaries for phrase-based decoding. In: Proceedings of HLT-NAACL 2010

Yamabana Kiyoshi et al (2003) A speech translation system with mobile wireless clients. In: Proceedings of ACL 2003

Yamada Kenji, Kevin Knight (2001) A syntax-based statistical translation model ACL-EACL-2001: 39th annual meeting [of the association for computational linguistics] and 10th conference of the European chapter (of ACL), Toulouse, 9–11 July 2001, pp 523–530

Yamada Kenji, Kevin Knight (2002) A decoder for syntax-based statistical MT. In: ACL-2002: 40th annual meeting of the association for computational linguistics, Philadelphia, July 2002, pp 303–310. (PDF, 788 KB)

Yaqin Yang, Nianwen Xue (2010) Chasing the ghost: recovering empty categories in the Chinese Treebank. In: Proceedings of COLING, Beijing

Yonggang Deng, Jia Xu, Yuqing Gao (2008) Phrase table training for precision and recall: what makes a good phrase and a good phrase pair? In: Proceedings of ACL, Columbus, pp 81–88

Zens Richard, Hermann Ney (2006) Discriminative reordering models for statistical machine translation. In: HLT-NAACL 2006: proceedings of the workshop on statistical machine translation, New York, June 2006, pp 55–63

Zhai Feifei, Jiajun Zhang Yu Zhou, Chengqing Zong (2011) Simple but effective approaches to improving tree-to-tree model, MTS

Zhang Y (2008) Structured language models for statistical machine translation. Ph.D. dissertation, Carnegie Mellon University

Zhang Licheng Fang, Peng Xu, Xiaoyun Wu (2011) Binarized forest to string translation. In: ACL-HLT 2011: proceedings of the 49th annual meeting of the association for computational linguistics, Portland, 19–24 June 2011, pp 835–845

Zhang Ying (Joy) http://projectile.sv.cmu.edu/research/public/talks/speechtranslation/sst-survey-joy.pdf

Zhang Ying, Stephan Vogel (2007) PanDoRA: a large- scale two-way statistical machine translation system for hand-held devices. In: Proceedings of MT Summit XI, Copenhagen, pp 10–14

Zhang Min, Hongfei Jiang, Ai Ti Aw, Jun Sun, Sheng Li, Chew Lim Tan (2007) A tree-to-tree alignment-based model for statistical machine translation. In: Proceedings of MT Summit XI, Copenhagen, 10–14 Sept 2007, pp 535–542

Zhang Min, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, Sheng Li (2008) A tree sequence alignment-based tree-to-tree translation model. In: Proceedings of the conference ACL-08: HLT. 46th annual meeting of the association for computational linguistics: human language technologies, The Ohio State University, Columbus, 15–20 June 2008, pp 559–567

Zhanyi Liu, Haifeng Wang, Hua Wu, Sheng Li (2010) Improving statistical machine translation with monolingual collocation. In: Proceedings of ACL 2010

Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, Yun Huang (2009) Improving statistical machine translation using domain bilingual multiword expressions. In: Proceedings of MWE 2009 (ACL-IJCNLP)

Zhongqiang Huang, Cmejrek M, Zhou B (2010) Soft syntactic constraint for hierarchical phrase-based translation using latent syntactic distributions. In: Proceedings of EMNLP10, Stroudsburg

Zollmann A, Venugopal A (2006) Syntax augmented machine translation via chart parsing. In: Proceedings of the workshop on statistical machine translation, New York, pp 138–141

Zong Chenqing, Bo Xu, Taiyi Huang (2002) Interactive chinese-to-english speech translation based on dialogue management. In: Proceedings of the workshop on speech-to-speech translation: algorithms and systems, pp 61–68

Zong Chengqing, Heyan Huang, Shuming Shi (2008) Application of machine translation during Olympic Games 2008. In: AMTA-2008. MT at work: proceedings of the eighth conference of the association for machine translation in the Americas, Waikiki, 21–25 Oct 2008, pp 470–479

Online MT Systems and Tools

- http://www.languageweaver.com
- http://translate.google.com
- http://www.microsofttranslator.com
- http://www.systran.co.uk/
- http://www.freetranslation.com
- https://apertium.svn.sourceforge.net/svnroot/apertium/branches/apertium-dixtools-paradigmlearning
- http://www.opentrad.com
- http://www.eitb24.com/en
- http://www.cunei.org
- http://www.undl.org/
- http://www.unlweb.net/wiki/index.php/Introduction_to_UNL
- http://speechtrans.com/#
- http://researcher.ibm.com/researcher/view_page.php?id=2323
- http://www.youtube.com/watch?v=Ex-NBO_w0zQ
- http://www.babylon.com/mac.html
- http://itunes.apple.com/us/app/voicetra-speech-to-speech/id383542155?mt=8
- http://text-to-speech.imtranslator.net/
- http://www.speech.sri.com/projects/translation/
- http://www.bbn.com/technology/speech/speech_to_speech_translation
- http://www.ustar-consortium.com/
- http://www.research.att.com/projects/Speech_Translation/index.html?fbid=0GMC-dWS68d
- http://www.gizmag.com/go/2686/
- http://www.quaero.org
- http://www.speech.cs.cmu.edu/
- http://www.loquendo.com/it/
- http://www.is.cs.cmu.edu/mie/janus.html
- http://www-01.ibm.com/software/websphere/products/mobilespeech/
- http://research.microsoft.com/en-us/groups/srg/default.aspx
- http://tldp.org/HOWTO/Speech-Recognition-HOWTO/software.html
- http://cmusphinx.sourceforge.net/
- http://htk.eng.cam.ac.uk/
- http://julius.sourceforge.jp/en_index.php
- http://www.simon-listens.org/index.php?id=122&L=1
- http://www.sdl.com/en/language-technology/products/automated-translation/
- http://logos-os.dfki.de/
- http://www.openmatrex.org/
- http://tool.statmt.org/
- http://www.apertium.org/
- www.limsi.fr/tlp
- www.informatik.kit.edu/interact
- www-i6.informatik.rwth-aachen.de
- www.vocapia.com
- http://www.darpa.mil/ipto/Programs/gale/index.htm
- http://www.darpa.mil/
- http://www.ldc.upenn.edu/
- https://wit3.fbk.eu/
- http://iwslt2012.org/
- http://iwslt2012.org/index.php/evaluation-campaign/ted-task

Part III
Empirical Studies of Natural Language and Mobility

Chapter 7
Natural Language Technology in Mobile Devices: Two Grounding Frameworks

Jerome R. Bellegarda

Abstract Natural language technology is assuming an ever-expanding role in smartphones and other mobile devices, as advances in software integration and efforts toward more personalization and context awareness have brought closer the long-standing vision of the ubiquitous intelligent personal assistant. Far beyond merely offering more options in terms of user interface, this trend has the potential to usher in a genuine paradigm shift in human-computer interaction. This contribution reviews the two major semantic interpretation frameworks underpinning this more anthropomorphic style of interaction. It then discusses the inherent complementarity in their respective advantages and drawbacks, and speculates on how they might combine in the near future to best mitigate any downside. This prognosis would amount to achieving the best of both worlds in next-generation mobile devices.

Introduction

In the past few years, smartphones and other mobile devices, such as electronic tablets and more generally a wide variety of handheld media appliances, have gained critical importance in the quest for ubiquitous computing and communications. Numerous computing solutions and communication services are now readily accessible, anytime and (almost) anywhere, by an ever-growing number of users. In fact, the rapid uptake of information access, processing, and distribution via emails, texts, social networking, tweets, etc. has caused mobile data traffic volumes to overtake voice traffic volumes. Recent figures obtained using Cisco Visual Networking Index (2012) show that global mobile data traffic in

J.R. Bellegarda, Ph.D. (✉)
Apple Distinguished Scientist – Human Language Technologies, Apple Inc.,
MS 37-2FMW – One Infinite Loop, Cupertino, CA 95014, USA
e-mail: jerome@apple.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_7,
© Springer Science+Business Media New York 2013

2011 (597 petabytes per month) was over eight times greater than the entire global Internet traffic in 2000 (75 petabytes per month).

In hindsight, such widespread adoption of mobile devices by the general public was greatly facilitated by the rapid deployment of touch input, navigation, and manipulation, which effectively enabled untrained users to intuitively operate a new generation of sophisticated digital tools. Consumers of all persuasions near-universally adopted touch as a natural modality, capable of handling multiple interaction constraints within a single surface-driven experience. In essence, touch emerged as a necessary affordance for the familiar graphical user interface to make sense in a mobile context.

This emphasis on touch may seem paradoxical in the case of the telephone, the prototypical speech-centric device. The growing precedence of data traffic over voice traffic, however, both accounts for and reflects the decision to initially anchor the mobile experience to the standard desktop experience. The associated interaction model, optimized for a graphical user interface, is based on the direct manipulation of individual data objects. Using voice to perform such manipulation is comparatively impoverished, because it has evolved to deal with higher-level notions in more integrative fashion. Thus, under this interaction model, voice is hard to integrate as a first-class member of the user interface.

On the other hand, direct manipulation has itself shown increasing signs of stress recently, due to the convergence of a number of factors: (i) constantly improving wireless connectivity, which has led to an unprecedented reliance on the web in everyday life, (ii) the ensuing growing variety and complexity of websites and mobile applications, which renders navigation more heterogeneous and therefore cognitively more demanding, and (iii) the predominant use of small(er) screens in the mobile space, which makes data objects harder to manipulate. At the same time, voice interaction has benefited from steady improvements in underlying speech technologies (largely driven by a greater quantity of labeled speech data to build better models), as well as the relative decrease in the cost of computing power necessary to implement comparatively more sophisticated solutions.

Not coincidentally, multiple voice-driven initiatives have now reached commercial deployment, with products like Apple's Siri (Apple 2011), Google's Voice Actions (Google Mobile 2008), Microsoft's Bing Voice Search (Microsoft Tellme 2008), Nuance's Dragon Go! (Nuance 2011), and Vlingo (Vlingo Mobile Voice User Interface 2008). The well-publicized release of Siri in Apple's iPhone 4S, in particular, may well have heralded a shift of the mainstream mobile user interface (back) toward a more speech-centric interaction model. This alternative is based on the "personal assistant" paradigm: just say what you want, and the system will figure out what the best course of action is. For example, to create a new entry on his/her calendar, the user may start the interaction with an input like:

Schedule a meeting with John Monday at 2 pm (7.1)

The system then has to recognize that the user's intent is to create a new entry, and deal with any ambiguities about the attributes of the entry, such as who will be

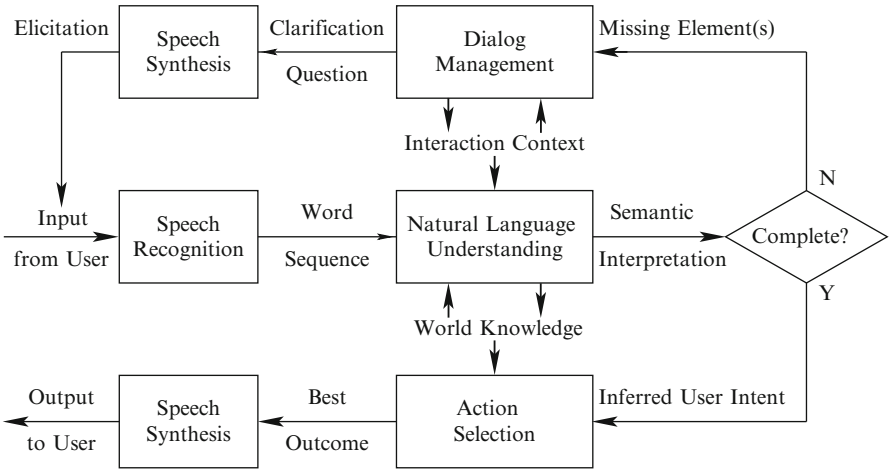

Fig. 7.1 Overview of "Personal Assistant" interaction model

invited (*John Smith* rather than *John Monday*), when the meeting will take place (*this coming Monday* rather than *last Monday*), etc.

Critical to this more anthropomorphic style of interaction is natural language technology, in its most inclusive definition encompassing speech recognition, speech synthesis, natural language understanding, and dialog management. An overview of this interaction model is given in Fig. 7.1. Note that success in this realm is measured in subjective terms: *how well* does the system fulfill the needs of the user relative to his/her intent and expectations? Depending on the task, "well" may variously translate into "efficiently" (with minimal interruption),"thoroughly" (so the task is truly complete) and/or "pleasantly" (as might have occurred with a human assistant).

Of course, many of the core building blocks shown in Fig. 7.1 have already been deployed in one form or another, for example in customer service applications with automatic call handling. Wildfire, a personal telephone assistant, has similarly been available since the mid 1990s (Wildfire Virtual Assistant Service 1995). Yet in most consumers' perception, at best the resulting interaction has not been significantly more satisfying than pressing touch-tone keys. So how to explain the widespread acceptance of Siri and similar systems? While the interaction model of Fig. 7.1 has not suddenly become flawless, it has clearly matured enough to offer greater perceived flexibility. Perhaps a key element of this perception is that the new systems strive to provide a direct answer whenever possible, rather than possibly heterogeneous information that may contain the answer.

Arguably, the most important ingredient of this new perspective is the accurate inference of user intent and correct resolution of any ambiguity in associated attributes. While speech input and output modules clearly influence the outcome by introducing uncertainty into the observed word sequence, the correct delineation of the task and thus its successful completion heavily hinges on the appropriate semantic interpretation of this sequence. This contribution accordingly focuses on the two

major frameworks that have been proposed to perform this interpretation, and reflects on how they can each contribute to a positive mobile experience.

The material is organized as follows. The next section describes the rule-based framework underpinning expert systems and similar ontology-based efforts, while section "Statistical Framework" does the same for the statistical framework characteristic of data-driven systems. In section "Toward a Cognitive Interaction Model", we discuss the inevitable convergence between these two historically separate frameworks, and how this is likely to further strengthen the cognitive aspects of natural language understanding. Finally, section "Conclusion" concludes the chapter with a summary of the material covered, and some prognostications regarding the role of mobile devices in the emergence of the next-generation user interaction model.

Rule-Based Framework

In this section we first give some historical perspectives on the rule-based framework, then review the current state-of-the-art, and finally discuss the major trade-offs associated with the approach.

Background

At its core, the rule-based framework draws its inspiration from early expert systems such as MYCIN (Buchanan and Shortliffe 1984). These systems, relying on an inference engine operating on a knowledge base of production rules, were firmly rooted in the artificial intelligence (AI) tradition (Laird et al. 1987). Their original purpose was to create specialized agents aimed at assisting humans in specific domains (cf., e.g., Morris et al. 2000). Agent frameworks were later developed to create personal intelligent assistants for information retrieval. In this context, the Open Agent Architecture (OAA) introduced the powerful concept of delegated computing (Cheyer and Martin 2001). This was later extended to multi-agent scenarios where distributed intelligent systems can model independent reactive behavior (cf., e.g., Sycara et al. 2001).

In the mid-2000s, DARPA's PAL (Perceptive Assistant that Learns) program attempted to channel the above efforts into a learning-based intelligent assistant comprising natural language user interaction components layered on top of core AI technologies such as reasoning, constraint solving, truth maintenance, reactive planning, and machine learning (Berry et al. 2005). The outcome, dubbed CALO for the Cognitive Assistant that Learns and Organizes, met the requirements for which it was designed, but because of its heterogeneity and complexity, it proved difficult for non-experts to leverage its architecture and capabilities across multiple domains. This sparked interest in a more streamlined design where user interaction, language processing and core reasoning are more deeply integrated within a single unified framework (Guzzoni et al. 2006).

An example of such framework is the "Active" platform, which eschews some of the sophisticated AI core processing in favor of a lighter-weight, developer-friendly version easier to implement and deploy (Guzzoni et al. 2006). An application based on this framework consists of a set of loosely coupled services interfacing with specialized task representations crafted by a human expert. Using loosely coupled services eases integration of sensors (cf. speech recognition, but also vision systems, mobile or remote user interfaces, etc.), effectors (cf. speech synthesis, but also touch user interfaces, robotics, etc.) and processing services (such as remote data sources and other processing components).

Current State-of-the-Art

In the "Active" framework, every task is associated with a specific "active ontology." Whereas a conventional ontology is a static data structure, defined as a formal representation for domain knowledge, with distinct classes, attributes, and relations among classes, an active ontology is a dynamic processing formalism where distinct processing elements are arranged according to ontology notions. An active ontology thus consists of a relational network of concepts, where concepts serve to define both data structures in the domain (e.g., a meeting has a date and time, a location, a topic and a list of attendees) as well as associated rule sets that perform actions within and among concepts (e.g., the date concept derives a canonical date object of the form: `date(DAY,MONTH,YEAR,HOURS,MINUTES)` from a word sequence such as *Monday at 2 pm*).

Rule sets are collections of rules where each rule consists of a condition and an action. As user input is processed, data and events are inserted into a fact store responsible for managing the life cycle of facts. Optional information can be specified to define when the fact should actually be asserted and when it should be removed. As soon as the contents of the fact store changes, an execution cycle is triggered and conditions evaluated. When a rule condition is validated, the associated action is executed. The active ontology can therefore be viewed as an execution environment.

To fix ideas, Fig. 7.2 shows the active ontology associated with the meeting scheduling task mentioned earlier. The active ontology consists of a tree-like structure defining the structure of a valid command for this task. The command operates on a complete event concept representing the action of scheduling a meeting. The meeting concept itself has a set of attributes comprising one or more persons, a topic, a location and a date. Structural relationships are denoted by arrows, which relate to a "has a" ontological notion. For instance, topic, date, location and person concepts are members of a meeting.

Structural relationships also carry cardinality information and record whether children nodes are optional, mandatory, unique or multiple. For instance, the relationship between person and meeting is multiple and mandatory, which is denoted by a double solid arrow. On the other hand, the relationship between topic and meeting is unique and optional, which is denoted by a single dashed

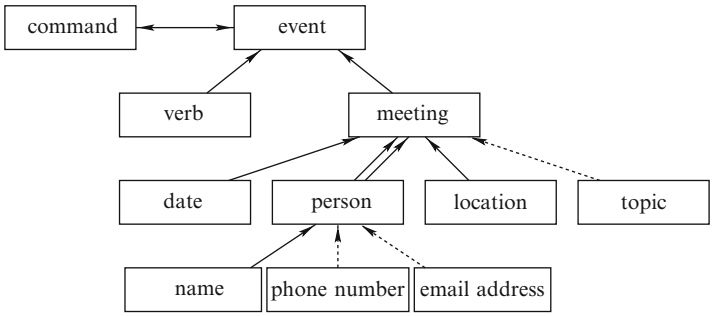

Fig. 7.2 Active ontology for meeting scheduling task

arrow. This structure is used to provide the user with contextual information. In the example above, as the location node is linked as mandatory, the user will be asked to provide a location. Through this mechanism, the active ontology not only generates a structured command but also builds dynamic information to interactively assist the user.

As alluded to earlier, concepts incorporate various instantiations of canonical objects. For example, *Monday at 2pm* and *tomorrow morning* are two instances of date objects in the date concept. These objects relate to a "is a" ontological notion. To the extent that rule sets can be specified to sense and rate incoming words about their possible relevance to various concepts, this makes the domain model portable across languages. In addition, it has the desirable side effect of making the approach insensitive to the order of component phrases.

Trade-Offs

An obvious bottleneck in the development of rule-based systems such as described above is the specification of active ontologies relevant to the domain at hand. For the system to be successful, each ontology must be 100 % complete: if an attribute is overlooked, or a relationship between classes is missing, some (possibly rare) user input will not be handled correctly. In practice, this requires the task domain to be sufficiently well-specified that a human expert from the relevant field is able to distill it into the rule base. This so-called knowledge engineering is typically hard to "get right" with tasks that are highly variable or subject to a lot of noise.

On the plus side, once the ontology correctly captures the whole domain structure, deployment across multiple languages is relatively straightforward. Since a near-exhaustive list of relevant word patterns is already included inside each concept, and word order is otherwise largely ignored, only individual surface forms have to translated. This makes this approach paradoxically similar in spirit to (data-driven) bag-of-words techniques such as latent semantic mapping (Bellegarda 2005).

Statistical Framework

As in the previous section, we first give some historical perspectives on the statistical framework, then review the current state-of-the-art, and finally discuss the major trade-offs associated with the approach.

Background

Pervasive in the earlier discussion of rule-based systems is the implicit assumption that language can be satisfactorily modeled as a finite state process. Strictly speaking, this can only be justified in limited circumstances, since, in general, the level of complexity of human languages goes far beyond that of context-free languages. This realization sparked interest in an alternative outlook not grounded in artificial intelligence, but rather in statistical modeling. This strand of work originated in speech recognition, where in the 1980s probabilistic models such as Hidden Markov Models were showing promise for reconstructing words from a noisy speech signal (Rabiner et al. 1996). Applying similar probabilistic methods to natural language understanding involved the integration of data-driven evidence gathered on suitable training data in order to infer the user's intent.

The theoretical underpinnings for this kind of reasoning were first developed in the context of a partially observable Markov decision process (POMDP) (Sondik 1971). The key features of the POMDP approach are (i) the maintenance of a system of beliefs, continually updated using Bayesian inference, and (ii) the use of a policy whose performance can be quantified by a system of associated rewards and optimized using reinforcement learning via Bellman's optimality principle (Kaelbling et al. 1998). Note that Bayesian belief tracking and reward-based reinforcement learning are mechanisms that humans themselves appear to use for planning under uncertainty (Fu and Anderson 2006). For example, experimental data shows that humans can implicitly assimilate Bayesian statistics and use Bayesian inference to solve sensorimotor problems (Kording and Wolpert 2004).

This in turn motivated the application of the POMDP framework to spoken dialog systems, to similarly learn statistical distributions by observation and use Bayes' rule to infer posteriors from these distributions (Williams et al. 2005). However, this proved challenging in practice for several reasons. First, the internal state is a complex combination of the user's goal, the user's input, and the dialog history, with significant uncertainty in the user's utterances (due to speech recognition errors) propagating uncertainty into the other entities as well. In addition, the system action space must cover every possible system response, so policies must map from complex and uncertain dialog states into a large space of possible actions.

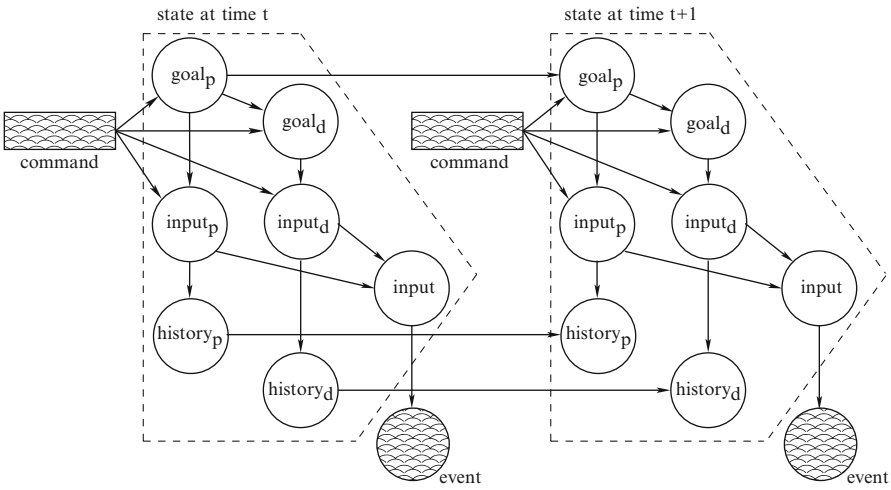

Fig. 7.3 (Partial) Dynamic Bayesian network for meeting scheduling task

Current State-of-the-Art

Making the POMDP framework tractable for real-world tasks typically involves a number of approximations. First, state values can be ranked and pruned to eliminate those not worth maintaining. Second, joint distributions can be factored by invoking some independence assumptions that can be variously justified from domain knowledge. Third, the original state space can be mapped into a more compact summary space small enough to conduct effective policy optimization therein. Fourth, in a similar way, a compact action set can be defined in summary space and then mapped back into the original master space (Williams and Young 2007).

As an example, Fig. 7.3 shows a possible POMDP implementation for the meeting scheduling task mentioned earlier. It illustrates one time step of a (partial) dynamic Bayesian network, in which the (hidden) system state and (observed) event are represented by open and shaded circles, respectively, while the (observed) command executed by the system is denoted by a shaded rectangle. The state is decomposed into slots representing features such as *person* (indexed by *p*), *date* (indexed by *d*), *location* and *topic* (not shown). Each slot comprises information related to user goal, user input, and dialog history so far. In this simple example, the only dependence modeled between slots is related to the person information. This configuration, known as a "Bayesian Update of Dialog State" (BUDS) system (Thomson et al. 2008), retains the ability to properly represent system dynamics and to use fully parametric models, at the cost of ignoring much of the conditional dependency inherent in real-world domains.

Because the state of the system (encapsulating the intent of the user) is a hidden variable, its value can only be inferred from knowledge of the transition probabilities

between two successive time instants and the observation probabilities associated with the observed event. This leads to a belief update equation of the form:

$$b_{t+1} = K \cdot O(o_{t+1}) \cdot T(c_t) \cdot b_t, \tag{7.2}$$

where the N-dimensional vector $b = [b(s_1) \ldots b(s_N)]^T$ is the belief distribution over N possible system states s_i, $O(o)$ is a diagonal matrix of observation probabilities $P(o|s_i)$, and $T(c)$ is the $N \times N$ transition matrix for command c. Given some assumed initial value b_0, (7.2) allows the belief state to be updated as each user input is observed. Since the actual state is unknown, the action taken at each turn must be based on the belief state rather than the underlying hidden state.

This mapping from belief state to action is determined by a policy $\pi : b \rightarrow c$. The quality of any particular policy is quantified by assigning rewards $r(s, c)$ to each possible state-command pair. The choice of specific rewards is a design decision typically dependent on the application. Different rewards will result in different policies and most likely divergent user experiences. However, once the rewards have been fixed, policy optimization is equivalent to maximizing the expected total reward over the course of the user interaction. Since the process is assumed to be Markovian, the total reward expected in traversing from any belief state b to the end of the interaction following policy π is independent of all preceding states. Using Bellman's optimality principle, it is possible to compute the optimal value of this value function iteratively. As mentioned earlier, this iterative optimization is an example of reinforcement learning (Sutton and Barto 1998).

Trade-Offs

From a theoretical perspective, the POMDP approach has many attractive properties: by integrating Bayesian belief monitoring and reward-based reinforcement learning, it provides a robust interpretation of imprecise and ambiguous human interactions, promotes the ability to plan interactions so as to maximize concrete objective functions, and offers a seamless way to encompass short-term adaptation and long term learning from experience within a single statistical framework. Still, it is potentially fragile when it comes to assigning rewards, as encouraging (respectively discouraging) the correct (respectively wrong) state-command pair can be a delicate exercise in the face of a huge space of possible such pairs.

In addition, as is clear from (7.2), the computational complexity of a single inference operation is $\mathcal{O}(N^2)$, where N is the number of possible system states. Thus, for even moderately large values of N exact computation becomes intractable, which makes it challenging to apply to real-world problems. The necessary approximations all have drawbacks, be it in terms of search errors, spurious independence assumptions, quantization loss from master to summary space, or imperfect user simulation to generate reinforcement data (Gasic et al. 2008).

Toward a Cognitive Interaction Model

In this section, we first discuss the inherent complementarity of the two frameworks described above, and then speculate on how they might combine to best mitigate any downside.

Complementarity

Perhaps because rule-based and statistical frameworks have evolved in historically separate ways, the trade-offs they involve appear to be complementary. This complementarity can be traced in part to the top-down vs. bottom-up strategies framing the discussion. Whereas the statistical framework calls for a large amount of suitable training data to be collected beforehand, rule-based systems can be deployed right away. On the other hand, ontology specification requires upfront labor-intensive human expertise, while data-driven systems can be run in completely automated fashion. On the flip side, the former is much more amenable to leveraging know-how across languages, thus enabling rapid deployment in multiple languages, while in the latter every language essentially involves the same amount of effort.

Complementarity between the frameworks, however, goes beyond a mere data-vs-knowledge distinction. Whereas generally rule-based systems tend to be sensitive to noise, in principle the POMDP approach can cope with various sources of uncertainty. Yet its elegant optimization foundation assumes suitable reward specifications, which is probably best informed by empirical observations, and thus rules derived therefrom. In addition, POMDP systems typically involve a number of possibly deleterious approximations to reduce the computational complexity inherent to the sophisticated mathematical machinery involved. In contrast, the AI framework require fewer simplifying assumptions and thus may exhibit a more predictable behavior.

Convergence

The complementarity just evoked at multiple levels bodes well for an eventual convergence between the two frameworks. At the moment, rule-based systems are substantially easier to deploy, partly due to the paucity of real data available to train statistical systems, as well as the difficulties inherent to data collection via user simulation. As more devices acquire voice-driven "personal assistant" capabilities, however, the amount of training data observed under real-world conditions will soon become staggering. Assuming that appropriate techniques are developed to harness this data for the purpose of statistical training,

one of the big limiting factors in properly handling uncertainty can potentially disappear. By enabling more robust reasoning and adaptation, this should in turn considerably strengthen the cognitive aspects of natural language understanding.

At that point, it will become possible for the mainstream mobile user interface to pervasively leverage the "personal assistant" model across many more usage scenarios. The user will increasingly get used to expressing a general need, and letting the system stochastically fulfill it. The associated interaction model, based on intelligent delegation, is especially compelling in a mobile context because of its generality, simplicity, and efficiency. In addition, it is the only truly viable solution in eyes-busy or hands-free situations, such as when driving a vehicle. It is therefore likely that in the future mobile devices will (continue to) provide major impetus toward such fully cognitive interaction.

Conclusion

In this contribution, we have examined the expanding role of natural language technology in mobile devices, particularly as it pertains to the emerging deployment of the "personal assistant" style of interaction. Under this model it is critical to accurately infer user intent, which in turn hinges on the appropriate interpretation of the words uttered. We have reviewed the two major frameworks within which to perform this interpretation, along with their most salient advantages and drawbacks. Ontology-based systems are better suited for initial deployment in well-defined domains across multiple languages, but must be carefully tuned for optimal performance. Data-driven systems have the potential to be more robust, as long as they are trained on enough quality data.

For the majority of users, adopting the "personal assistant" interaction model means a chance to intuitively operate more and more sophisticated digital tools, but also a switch to an unfamiliar user interface paradigm. Thirty years ago, the graphical user interface had a revolutionary impact in driving growth in acceptance and usability of personal computers. Whether the personal assistant interface, mobile or otherwise, ultimately has a similar impact on the industry remains to be seen. But what is certain is this interaction model will be increasingly available alongside, and in all likelihood tightly integrated with, the traditional object manipulation model.

As this style of interaction becomes truly mainstream, we can in turn expect a considerable amount of both qualitative user feedback and quantitative real-world data to inform the next-generation mobile experience. We therefore see it as inevitable that the two historically separate frameworks reviewed in this contribution will converge, and thereby bring a new level of cognition to natural language understanding. The perspective of such convergence bodes well for the continued leadership role of mobile devices going forward. The cognitive interaction model they ultimately call for will likely be a key stepping stone toward an ever more tangible vision of ubiquitous intelligence.

References

Apple Inc. (2011) http://www.apple.com/iphone/features/siri.html
Bellegarda JR (2005) Latent semantic mapping. In: Deng L, Wang K, Chou W (eds) Signal Proc Mag 22(5):70–80. Special issue Speech Technol Syst Human–Machine Communication
Berry P, Myers K, Uribe T, Yorke-Smith N (2005) Constraint solving experience with the CALO project. In: Proceedings of the workshop constraint solving under change and uncertainty, Sitges, pp 4–8
Buchanan BG, Shortliffe EH (1984) Rule–based expert systems: the MYCIN experiments of the stanford heuristic programming project. Addison–Wesley, Reading
Cisco Visual Networking Index (2012) Global mobile data traffic forecast. Update, 2011–2016, http://www.cisco.com/en/US/netsol/ns827/networking_solutions_white_papers_list.html
Cheyer A, Martin D (2001) The open agent architecture. J Auton Agent Multi Agent Syst 4(1):143–148
Fu W-T, Anderson J (2006) From Recurrent choice to skill learning: a reinforcement-learning model. J Exp Psychol Gen 135(2):184–206
Gasic M, Keizer S, Mairesse F, Schatzmann J, Thomson B, Young S (2008) Training and evaluation of the HIS POMDP dialogue system in noise. In: Proceedings of the 9th SIGdial workshop discourse dialog, Columbus
Google Mobile (2008) http://www.google.com/mobile/voice-actions
Guzzoni D, Baur C, Cheyer A (2006) Active: a unified platform for building intelligent web interaction assistants. In: Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, Hong Kong
Kaelbling JL, Littman M, Cassandra A (1998) Planning and acting in partially observable stochastic domains. Artif Intell 101:99–134
Kording JK, Wolpert D (2004) Bayesian integration in sensorimotor learning. Nature 427:224–227
Laird JE, Newell A, Rosenbloom PS (1987) SOAR: an architecture for general intelligence. Artif Intell 33(1):1–64
Microsoft Tellme (2008) http://www.microsoft.com/en-us/Tellme/consumers/default.aspx
Morris J, Ree P, Maes P (2000) SARDINE: dynamic seller strategies in an auction marketplace. In Proceedings of the ACM conference electronic commerce, New York, pp 128–134
Nuance Dragon Go! (2011) http://www.nuance.com/products/dragon-go-in-action/index.htm
Rabiner LR, Juang BH, Lee C-H (1996) An overview of automatic speech recognition. In: Lee C-H, Soong FK, and Paliwal KK (eds) Automatic speech and speaker recognition: advanced topics, chapter 1. Kluwer Academic, Boston, pp 1–30
Sondik E (1971) The optimal control of partially observable markov decision processes. Ph.D. Dissertation, Stanford University, Palo Alto
Sutton R, Barto A (1998) Reinforcement learning: an introduction. Series on adaptive computation and machine Learning. MIT, Cambridge
Sycara K, Paolucci M, van Velsen M, Giampapa J (2001) The RETSINA MAS infrastructure. Technical Report CMU- RI-TR-01-05, Robotics Institute Technical Report, Carnegie Mellon
Thomson B, Schatzmann J, Young S (2008) Bayesian update of dialogue state for robust dialogue systems. In: Proceedings of the international conference acoustics speech signal processing, Las Vegas
Vlingo Mobile Voice User Interface (2008) http://www.vlingo.com/
Wildfire Virtual Assistant Service (1995) Virtuosity Corp., http://www.wildfirevirtualassistant.com
Williams J, Young S (2007) Scaling POMDPs for spoken dialog management. IEEE Trans Audio Speech Lang Process 15(7):2116–2129
Williams J, Poupart P, Young S (2005) Factored partially observable markov decision processes for dialogue management. In: Proceedings of the 4th workshop knowledge reasoning in practical dialogue systems, Edinburgh

Chapter 8
Empirical Exploration of Language Modeling for the google.com Query Stream as Applied to Mobile Voice Search

Ciprian Chelba and Johan Schalkwyk

Abstract Mobile is poised to become the predominant platform over which people access the World Wide Web. Recent developments in speech recognition and understanding, backed by high bandwidth coverage and high quality speech signal acquisition on smartphones and tablets are presenting the users with the choice of speaking their web search queries instead of typing them. A critical component of a speech recognition system targeting web search is the language model. The chapter presents an empirical exploration of the google.com query stream with the end goal of high quality statistical language modeling for mobile voice search. Our experiments show that after text normalization the query stream is not as "wild" as it seems at first sight. One can achieve out-of-vocabulary rates below 1% using a 1 million word vocabulary, and excellent n-gram hit ratios of 77/88% even at high orders such as $n = 5/4$, respectively. A more careful analysis shows that a significantly larger vocabulary (approx. 10 million words) may be required to guarantee at most 1% out-of-vocabulary rate for a large percentage (95%) of users. Using large scale, distributed language models can improve performance significantly—up to 10% relative reductions in word-error-rate over conventional models used in speech recognition. We also find that the query stream is non-stationary, which means that adding more past training data beyond a certain point provides diminishing returns, and may even degrade performance slightly. Perhaps less surprisingly, we have shown that locale matters significantly for English query

C. Chelba, Ph.D. (✉)
Staff Research Scientist, Google, Inc.,
1600 Amphiteatre Parkway, Mountain View, CA 94043, USA
e-mail: ciprianchelba@google.com

J. Schalkwyk, M.Sc.
Principal Staff Engineer, Google, Inc., 76 Ninth Avenue, 4th Floor, New York, NY 10011, USA

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_8,
© Springer Science+Business Media New York 2013

data across USA, Great Britain and Australia. In an attempt to leverage the speech data in voice search logs, we successfully build large-scale discriminative N-gram language models and derive small but significant gains in recognition performance.

Introduction

Mobile web search is a rapidly growing area of interest. Internet-enabled smartphones account for an increasing share of 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 searching the web, 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. To address these problems, Google introduced voice search in November 2008. The goal of Google voice search is to recognize any spoken search query, and be capable of handling anything that Google search can handle.

We present an empirical exploration of `google.com` query stream language modeling for voice search. We describe the normalization of the typed query stream resulting in out-of-vocabulary (OOV) rates below 1% for a 1 million word vocabulary. We present a comprehensive set of experiments that guided the design decisions for a voice search service. In the process we re-discovered a less known interaction between Kneser-Ney smoothing and entropy pruning, and found empirical evidence that hints at non-stationarity of the query stream, as well as strong dependence on various English locales—USA, Britain and Australia.

In an attempt to leverage the large amount of speech data made available by the voice search service, we present a distributed framework for large-scale discriminative language models that can be integrated within a large vocabulary continuous speech recognition (LVCSR) system using lattice rescoring. We intentionally use a weakened acoustic model in a baseline LVCSR system to generate candidate hypotheses for voice search data; this allows us to utilize large amounts of unsupervised data to train our models. We propose an efficient and scalable MapReduce framework that uses a perceptron-style distributed training strategy to handle these large amounts of data. We report small but significant improvements in recognition accuracies on a standard voice search data set using our discriminative reranking model. We also provide an analysis of the various parameters of our models including model size, types of features, size of partitions in the MapReduce framework with the help of supporting experiments.

We will begin by defining the language modeling problem and typical metrics for comparing language models. We will then describe a series of experiments which explore the dimensions along which Voice Search language models may be refined.

Language Modeling Basics

A statistical language model estimates the prior probability values $P(W)$ for strings of words W in a vocabulary $\mathcal{V}$ whose size is usually in the tens or hundreds of thousands. Typically the string W is broken into sentences, or other segments such as utterances in automatic speech recognition, which are assumed to be conditionally independent. For the rest of this chapter, we will assume that W is such a segment, or sentence. With $W = w_1, w_2, \ldots, w_n$ we get:

$$P(W) = \prod_{i=1}^{n} P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \tag{8.1}$$

Since the parameter space of $P(w_k \mid w_1, w_2, \ldots, w_{k-1})$ is too large, the language model is forced to put the *context* $W_{k-1} = w_1, w_2, \ldots, w_{k-1}$ into an *equivalence class* determined by a function $\Phi(W_{k-1})$. As a result,

$$P(W) \cong \prod_{k=1}^{n} P(w_k \mid \Phi(W_{k-1})) \tag{8.2}$$

Research in language modeling consists of finding appropriate equivalence classifiers Φ and methods to estimate $P(w_k \mid \Phi(W_{k-1}))$.

The most successful paradigm in language modeling uses the *(n − 1)-gram* equivalence classification, that is, defines

$$\Phi(W_{k-1}) \doteq w_{k-n+1}, w_{k-n+2}, \ldots, w_{k-1}$$

Once the form $\Phi(W_{k-1})$ is specified, only the problem of estimating $P(w_k \mid \Phi(W_{k-1}))$ from training data remains. In most practical cases, $n = 3$ which leads to a *trigram* language model.

Perplexity as a Measure of Language Model Quality

A *statistical language model* can be evaluated by how well it predicts a string of symbols W_t—commonly referred to as *test data*—generated by the source to be modeled.

Assume we compare two models M_1 and M_2 using the same vocabulary[1] $\mathcal{V}$. They assign probability $P_{M_1}(W_t)$ and $P_{M_2}(W_t)$, respectively, to the sample test string W_t. The test string has neither been used nor seen at the estimation step

[1] Language models estimated on different vocabularies cannot be directly compared using perplexity, since they model completely different probability distributions.

of either model and it was generated by the same source that we are trying to model. "Naturally", we consider M_1 to be a better model than M_2 if $P_{M_1}(W_t) > P_{M_2}(W_t)$.

A commonly used quality measure for a given model M is related to the entropy of the underlying source and was introduced under the name of *perplexity* (PPL) (Jelinek 1997):

$$PPL(M) = exp(-\frac{1}{N}\sum_{k=1}^{N} \ln[P_M(w_k \mid W_{k-1})]) \tag{8.3}$$

To give intuitive meaning to perplexity, it represents the number of guesses the model needs to make in order to ascertain the identity of the next word, when running over the test word string from left to right. It can be easily shown that the perplexity of a language model that uses the uniform probability distribution over words in the vocabulary $\mathcal{V}$ equals the size of the vocabulary; a good language model should of course have lower perplexity, and thus the vocabulary size is an upper bound on the perplexity of a given language model.

Very likely, not all words in the test string W_t are part of the language model vocabulary. It is common practice to map all words that are out-of-vocabulary to a distinguished *unknown word* symbol, and report the out-of-vocabulary (OOV) rate on test data—the rate at which one encounters OOV words in the test string W_t—as yet another language model performance metric besides perplexity. Usually the unknown word is assumed to be part of the language model vocabulary—*open vocabulary* language models—and its occurrences are counted in the language model perplexity calculation, Eq. (8.3). A situation far less common in practice is that of *closed vocabulary* language models where all words in the test data will always be part of the vocabulary $\mathcal{V}$.

Smoothing

Since the language model is meant to assign non-zero probability to unseen strings of words (or equivalently, ensure that the cross-entropy of the model over an arbitrary test string is not infinite), a desirable property is that:

$$P(w_k \mid \Phi(W_{k-1})) > \epsilon > 0, \forall w_k, W_{k-1}, \tag{8.4}$$

also known as the *smoothing* requirement.

A large body of work has accumulated over the years on various smoothing methods for n-gram language models that ensure this to be true. The two most widespread smoothing techniques are probably Kneser-Ney (1995) and Katz (1987); Goodman (2001) provides an excellent overview that is highly recommended to any practitioner of language modeling.

Query Language Modeling for Voice Search

A typical voice search language model used in our system for the US English query stream is trained as follows:

- Vocabulary size: 1M words, OOV rate 0.57%
- Training data: 230B words, a random sample of anonymized queries from google.com that did not trigger spelling correction

The resulting size, as well as its performance on unseen query data (10k queries) when using Katz smoothing is shown in Table 8.1. We note a few key aspects:

- The first pass LM (15 million n-grams) requires very aggressive pruning—to about 0.1% of its unpruned size—in order to make it usable in static FST-based (Finite State Transducer-based) ASR decoders (Automatic Speech Recognition decoders)
- The perplexity hit taken by pruning the LM is significant, 50% relative; similarly, the 3-g hit ratio is halved
- The impact on WER due to pruning is significant, yet lower in relative terms—10% relative, as we show in section "Effect of Language Model Size on Speech Recognition Accuracy"
- The unpruned model has excellent n-gram hit ratios on unseen test data: 77% for $n=5$, and 97% for $n=3$
- The choice of $n=5$ is because using higher n-gram orders yields diminishing returns: a 7-g LM is 4 times larger than the 5-g LM trained from the same data and using the same vocabulary, at no gain in perplexity.

For estimating language models at this scale we have used the distributed language modeling tools built for statistical machine translation (Brants and Xu 2009; Brants et al. 2007) based on the MapReduce infrastructure described in section "Language Modeling Basics". Pruned language models used in the first pass of the ASR decoder are converted to ARPA (Paul and Baker 1992) and/or FST (Allauzen et al. 2007) format using an additional MapReduce pass with a single reducer, which can optionally apply the language model compression techniques described in Harb et al. (2009).

The next section describes the text normalization that allows us to use a 1 million word vocabulary and obtain out-of-vocabulary (OOV) rates lower than 1%, as well as the excellent n-gram hit ratios presented in Table 8.1.

We then present experiments that show the temporal and spatial dependence of the English language models. Somewhat unexpectedly, using more training data does not result in an improved language model despite the fact that it is extremely well matched to the unseen test data. Additionally, the English language models built from training data originating in three locales (USA, Britain, and Australia) exhibit strong locale-specific behavior, both in terms of perplexity and OOV rate.

We will then present speech recognition experiments on a voice search test set.

Table 8.1 Typical voice search 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

Privacy Considerations

Before delving into the technical aspects of our work, we wish to clarify the privacy aspects of our work with respect to handling user data.

All of the query data used for training, and testing models is strictly anonymous; the queries bear no user-identifying information. The only data saved after training are vocabularies, or n-gram counts. When working with session data, such as the experiments reported in section "Optimal Size, Freshness and Time-Frame for Voice Search Vocabulary", we are even stricter: the evaluation on test data is done by counting on streamed filtered query logs, without saving any data.

Text Normalization

In order to build a language model for spoken query recognition we boot-strap from written queries to `google.com`. Written queries provide a data-rich environment for modeling of queries. This requires robustly transforming written text into spoken form.

Table 8.2 lists a couple of example queries and their corresponding spoken equivalents. 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 approach to text normalization. Let $T(written)$ be an acceptor that represents the written query. Conceptually the spoken form is computed as follows

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

where $N(spoken)$ represents the transduction from written to spoken form. Note that composition with $N(spoken)$ 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.

Table 8.2 Example written queries and their corresponding spoken form

Written query	Spoken query
weather scarsdale, ny	weather scarsdale new york
	weather in scarsdale new york
bankofamerica.com	bank of america dot com
81 walker rd	eighty one walker road
10:30am	ten thirty A M
at&t	A T and T
espn	E S P N

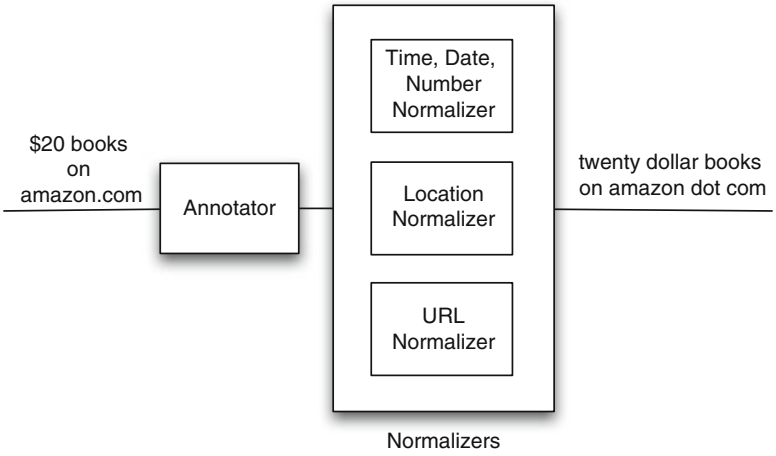

Fig. 8.1 Block diagram for context aware text normalization

The text normalization is run in multiple phases. Figure 8.1 depicts the text normalization process. In the first step we annotate the data. In this phase we categorize parts (sub strings) of queries into a set of known categories (e.g. time, date, url, location).

Since the query is annotated, it is possible to perform context-aware normalization on the substrings. Each category has a corresponding text normalization transducer $N_{cat}(spoken)$ 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 $N(spoken)$ using context dependent rewrite rules. For the URL normalizer $N_{url}(spoken)$ we train a statistical word decompounder that segments the string into its word constituents. For example, one reads the URL cancercentersofamerica.com as "cancer centers of america dot com". The URL decompounding transducer (decompounder) is built from the annotated data. Let Q

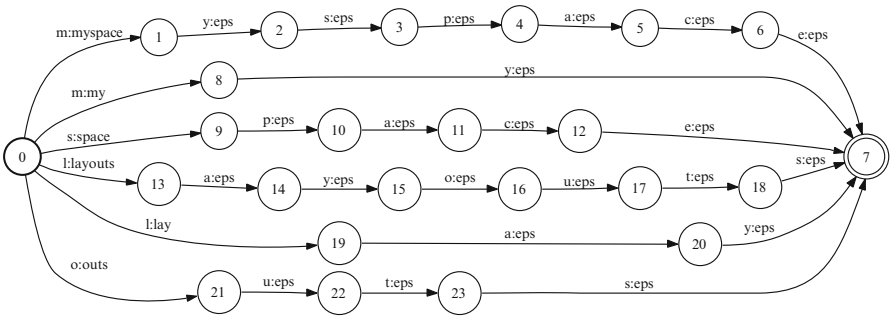

Fig. 8.2 $T(S)$ for the set of words $S=\{my, space, myspace, lay, outs, layouts\}$ where '{ eps}' denotes ε

be the set of queries in this Table, and let U be the set of substrings of these queries that are labeled URLs.

For a string s of length k let $I(s)$ be the transducer that maps each character in s to itself; i.e., the i-th transition in $I(s)$ has input and output label $s(i)$. $I(s)$ represents the word segmented into characters. Further, let $T(s)$ be the transducer that maps the sequence of characters in s to s; i.e., the first transition in $T(s)$ has input $s(1)$ and output s, and the i-th transition, where $i \neq 1$, has input $s(i)$ and output ε. $T(s)$ represents the transduction of the spelled form of the word to the word itself. For a set of strings S, we define

$$T(S) = \bigoplus_{s \in S} T(s)$$

where $\oplus$ is the union operation on transducers. $T(S)$ therefore represents the transduction of the spelling of the word to the word itself for the whole vocabulary. Figure 8.2 illustrates the operation of $T(\cdot)$.

The queries in Q and their frequencies are used to train an LM L_{BASE}. Let V_{BASE} be its vocabulary. We build the decompounder as follows:

1. For each $u \in U$, define $N(u)$ as,

$$N(u) = \text{bestpath}\left(I(u) \circ T^*(V_{\text{BASE}}) \circ L_{\text{BASE}}\right) \tag{8.5}$$

   where '*' is the Kleene Closure, and '$\circ$' is the composition operator.
2. $N(U) = \bigoplus_{u \in U} N(u)$ is the URL decompounder.

The transducer $I(u) \circ T^*(V_{\text{BASE}})$ in (8.5) represents the lattice of all possible segmentations of u using the words in V_{BASE}, where each path from the start state to a final state in the transducer is a valid segmentation. The composition with the LM L_{BASE} scores every path. Finally, $N(u)$ is the path with the highest probability; i.e. the most likely segmentation.

As an example, Fig. 8.3 depicts $I(u) \circ T^*(V_{\text{BASE}})$ for $u = myspacelayouts$. Each path in this lattice is a valid decomposition, and in Table 8.3 we list a sample of

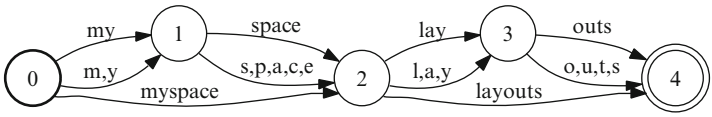

Fig. 8.3 The lattice $I(u) \circ T^*(V_{\text{BASE}})$ of all possible segmentations for $u = myspacelayouts$ using words in V_{BASE}

Table 8.3 Sample segmentations from Fig. 8.3. The one in bold represents the highest probability path as determined by the composition with L_{BASE}

Possible segmentations
myspace layouts
my space layouts
my space lay outs
my space l a y outs

these paths. After scoring all the paths via the composition with L_{BASE}, we choose the best path to represent the spoken form of the URL.

Language Model Refinement

Query Stream Non-stationarity

Our first attempt at improving the language model was to use more training data: we used a significantly larger amount of training data (`BIG`) vs. the most recent 230 billion (`230B`) prior to September 2008. The `230B` corpus is the most recent subset of `BIG`. As test data we used a random sample consisting of 10k queries from Sept to Dec 2008.

The first somewhat surprising finding was that this had very little impact in OOV rate for 1M word vocabulary: 0.77% (`230B` vocabulary) vs. 0.73% (`BIG` vocabulary). Perhaps even more surprising however is the fact that the significantly larger training set did not yield a better language model, despite the training data being clearly well matched, as illustrated in Table 8.4. In fact, we observed a significant reduction in PPL (10%) when using the more recent `230B` data. Pruning masks this effect, and the differences in PPL and WER become insignificant after reducing the language model size to approximately 10 million 3-g.

Since the vocabulary, and training data set change between the two rows, the PPL differences need to be analyzed in a more careful experimental setup.

Table 8.4 Pruned and unpruned 3-g language model perplexity when trained on the most recent 230 billion words, and a much larger amount of training data prior to test data, respectively

Training set	Test	Set PPL
	Unpruned	Pruned
230B	121	205
BIG	132	209

A superficial interpretation of the results seems to contradict the "there's no data like more data" dictum, recently reiterated in a somewhat stronger form in Banko and Brill (2001), Och (2005) and Halevy et al. (2009).

Our experience has been that supply of "more data" needs to be matched with increased demand on the modeling side, usually by increasing the model capacity—typically achieved by estimating more parameters. Experiments reported in section "Effect of Language Model Size on Speech Recognition Accuracy" improve performance by *keeping the amount of training data constant* (albeit very large), and *increasing the n-gram model size* by adding more *n*-grams at fixed *n*, as well as increasing the model order *n*. As such, it may well be the case that the increase in PPL for the BIG model is in fact due to limited capacity in the 3-g model.

More investigation is needed to disentangle the effects of query stream non-stationarity from possible mismatched model capacity issues. A complete set of experiments needs to:

- Let the *n*-gram order grow as large as the data allows;
- Build a sequence of models trained on exactly the same amount of data obtained by sliding a time-window of varying length over the query stream, and control for the ensuing vocabulary mismatches.

Effect of Language Model Size on Speech Recognition Accuracy

The work described in Harb et al. (2009) and Allauzen et al. (2009) enables us to evaluate relatively large query language models in the 1st pass of our ASR decoder by representing the language model in the OpenFst (Allauzen et al. 2007) framework. Figures 8.4 and 8.5 show the PPL and word error rate (WER) for two language models (3- and 5-g, respectively) built on the 230B training data, after entropy pruning to various sizes in the range 15 million–1.5 billion n-grams. Perplexity is evaluated on the test set described in section "Query Stream Non-stationarity"; word error rate is measured on another test set representative for the voice search task.

As can be seen, perplexity is very well correlated with WER, and the size of the language model has a significant impact on speech recognition accuracy: increasing the model size by two orders of magnitude reduces the WER by 10% relative.

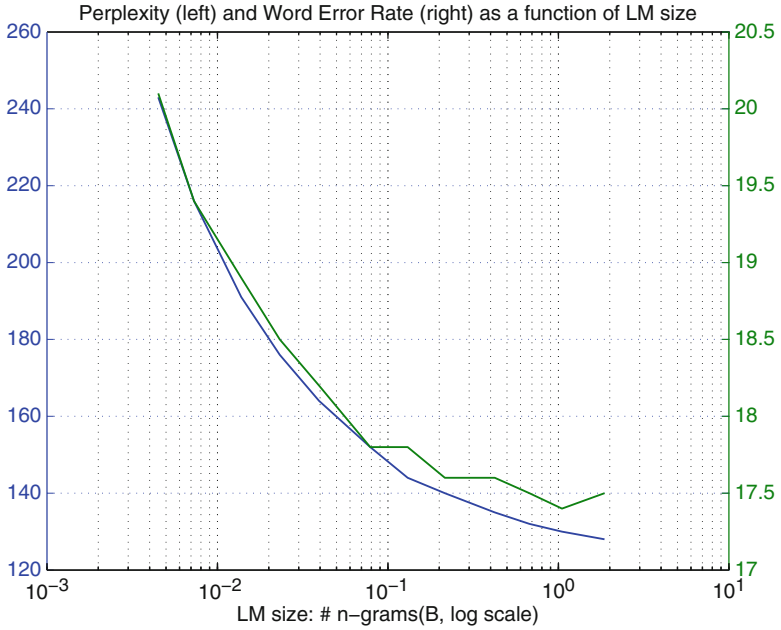

Fig. 8.4 Three-gram language model perplexity and word error rate as a function of language model size; *lower curve* is PPL

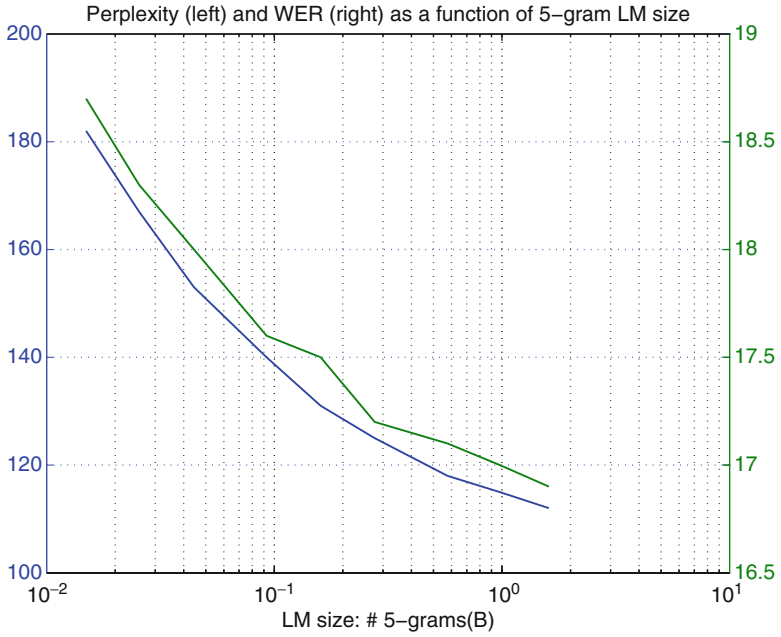

Fig. 8.5 Five-gram language model perplexity and word error rate as a function of language model size; *lower curve* is PPL

Table 8.5 Speech recognition language model performance when used in the 1st pass or in the 2nd pass—lattice rescoring

Pass	Language model	PPL	WER
1st	15M 3-g	191	18.7
1st	1.6B 5-g	112	16.9
2nd	15M 3-g	191	18.8
2nd	1.6B 5-g	112	16.9
2nd	12.7B 5-g	108	16.8

Table 8.6 Out of vocabulary rate: locale specific vocabulary halves the OOV rate

Training		Test	Locale
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**

We have also implemented lattice rescoring using the distributed language model architecture described in Brants et al. (2007), see the results presented in Table 8.5. This enables us to validate empirically the hypothesis that rescoring lattices generated with a relatively small first pass language model (in this case 15 million 3-g, denoted 15M 3-g in Table 8.5) yields the same results as 1st pass decoding with a large language model. A secondary benefit of the lattice rescoring setup is that one can evaluate the ASR performance of much larger language models.

Locale Matters

We also built locale specific English language models using training data prior to September 2008 across 3 English locales: USA (USA), Britain (GBR, about a quarter of the USA amount) and Australia (AUS, about a quarter of the GBR amount). The test data consisted of 10k queries for each locale, sampled randomly from Sept to Dec 2008.

Tables 8.6–8.8 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, either pruned or unpruned.

We have also built a combined model by pooling data across locales, with the results shown on the last row of Table 8.8. Combining the data negatively impacts all locales, in particular the ones with less data. 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 lumping all the data together, relative to using only the data from that given locale.

Table 8.7 Perplexity of unpruned LM: locale specific LM halves the PPL of the unpruned LM

Training		Test	Locale
Locale	USA	GBR	AUS
USA	**132**	234	251
GBR	260	**110**	224
AUS	276	210	**124**

Table 8.8 Perplexity of pruned LM: locale specific LM halves the PPL of the unpruned LM. Pooling all data is suboptimal

Training		Test	Locale
Locale	USA	GBR	AUS
USA	**210**	369	412
GBR	442	**150**	342
AUS	422	293	**171**
combined	227	210	271

The Case for Discriminative Language Modeling

The language model is a critical component of an automatic speech recognition (ASR) system that assigns probabilities or scores to word sequences. It is typically derived from a large corpus of text via maximum likelihood estimation in conjunction with some smoothing constraints. N-gram models have become the most dominant form of LMs in most ASR systems. Although these models are robust, scalable and easy to build, we illustrate a limitation with the following example from voice search. We expect a low probability for an ungrammatical or implausible word sequence. However, for a trigram like "a navigate to", a backoff trigram LM gives a fairly large LM log probability of -0.266 because both "a" and "navigate to" are popular words in voice search! Discriminative language models (DLMs) attempt to directly optimize error rate by rewarding features that appear in low error hypotheses and penalizing features in misrecognized hypotheses. In such a model, the trigram "a navigate to" receives a negative weight of -6.5 thus decreasing its chances of appearing as an ASR output. There have been numerous approaches towards estimating DLMs for large vocabulary continuous speech recognition (LVCSR) (Gao et al. 2005; Roark et al. 2007; Zhou et al. 2006).

There are two central issues that we discuss regarding DLMs. Firstly, DLM training requires large amounts of parallel data (in the form of correct transcripts and candidate hypotheses output by an ASR system) to be able to effectively compete

with n-gram LMs trained on large amounts of text. This data could be simulated using voice search logs from a baseline ASR system that are filtered by confidence score to obtain reference transcripts. However, this data is perfectly discriminated by first pass features such as the acoustic and language model scores, and leaves little room for learning. We propose a novel training strategy using lattices generated with a weaker acoustic model (henceforth referred to as *weakAM*) than the one used to generate reference transcripts for the unsupervised parallel data (referred to as the *strongAM*). This provides us with enough errors to derive large numbers of potentially useful word features; it is akin to using a weak LM in discriminative acoustic modeling to give more room for diversity in the word lattices resulting in better generalization (Schlüter et al. 1999). We conduct experiments to verify whether *weakAM*-trained models provide performance gains on rescoring lattices from a standard test set generated using *strongAM* (discussed in section "Evaluating ASR Performance on *v-search-test* Using DLM Rescoring").

The second issue is that discriminative estimation of LMs is computationally more intensive than regular N-gram LM estimation. The advent of distributed learning algorithms (Hall et al. 2010; Mann et al. 2009; McDonald et al. 2010) and supporting parallel computing infrastructure like MapReduce (Ghemawat and Dean 2004) has made it feasible to use large amounts of parallel data for training DLMs. We implement a distributed training strategy for the perceptron algorithm introduced by McDonald et al. (2010) using the MapReduce framework. Our design choices for the MapReduce implementation are specified in section "MapReduce Implementation Details" along with its modular nature thus enabling us to experiment with different variants of the distributed structured perceptron algorithm. Some of the descriptions in this paper have been adapted from previous work Jyothi et al. (2012).

The Distributed DLM Framework: Training and Implementation Details

Learning Algorithm

We aim to allow the estimation of large scale distributed models, similar in size to the ones in Brants et al. (2007). To this end, we make use of a distributed training strategy for the structured perceptron to train our DLMs (McDonald et al. 2010). Our model consists of a high-dimensional feature vector function Φ that maps an (utterance, hypothesis) pair (x, y) to a vector in R^d, and a vector of model parameters, $\mathbf{w} \in R^d$. Our goal is to find model parameters such that given x, and a set of candidate hypotheses $\mathcal{Y}$ (typically, as a word lattice or an N-best list that is obtained from a first pass recognizer), $\text{argmax}_{y \in \mathcal{Y}} \mathbf{w} \cdot \Phi(x, y)$ would be the $y \in \mathcal{Y}$ that minimizes the error rate between y and the correct hypothesis for x. For our experiments, the feature vector $\Phi(x, y)$ consists of AM and LM costs for y from the lattice $\mathcal{Y}$ for x, as

well as "word level n-gram features" which count the number of times different N-grams (of order up to 5 in our experiments) occur in y.

In principle, such a model can be trained using the conventional structured perceptron algorithm (Collins 2002). This is an online learning algorithm which continually updates $\mathbf{w}$ as it processes the training instances one at a time, over multiple training epochs. Given a training utterance $\{x_i, y_i\}$ ($y_i \in \mathcal{Y}_i$ has the lowest error rate with respect to the reference transcription for x_i, among all hypotheses in the lattice $\mathcal{Y}_i$ for x_i), if $\tilde{y}_i^* := \text{argmax}_{y \in \mathcal{Y}_i} \mathbf{w} \cdot \Phi(x_i, y)$ is not y_i, then $\mathbf{w}$ is updated to increase the weights corresponding to features in y_i and decrease the weights of features in $\tilde{y}_i^*$. During evaluation, we use parameters averaged over all utterances and over all training epochs. This was shown to give substantial improvements in previous work Collins (2002) and Roark et al. (2007).

Unfortunately, the conventional perceptron algorithm takes impractically long for the amount of training examples we have. We make use of a distributed training strategy for the structured perceptron that was first introduced in McDonald et al. (2010). The iterative parameter mixing strategy used in this paradigm can be explained as follows: the training data $\mathcal{T} = \{x_i, y_i\}_{i=1}^{N}$ is suitably partitioned into $\mathcal{C}$ disjoint sets $\mathcal{T}_1, \ldots, \mathcal{T}_{\mathcal{C}}$. Then, a structured perceptron model is trained on each data set in parallel. After one training epoch, the parameters in the $\mathcal{C}$ sets are mixed together (using a "mixture coefficient" μ_i for each set $\mathcal{T}_i$) and returned to each perceptron model for the next training epoch where the parameter vector is initialized with these new mixed weights. This is formally described in Algorithm 1; we call it "Distributed Perceptron". We also experiment with two other variants of distributed perceptron training, "Naive Distributed Perceptron" and "Averaged Distributed Perceptron". These models easily lend themselves to implementations using the distributed infrastructure provided by the MapReduce framework. The following section describes this infrastructure in greater detail.

MapReduce Implementation Details

We propose a distributed infrastructure using MapReduce (Ghemawat and Dean 2004) to train our large-scale DLMs on terabytes of data. The MapReduce (Ghemawat and Dean 2004) paradigm, adapted from a specialized functional programming construct, is specialized for use over clusters with a large number of nodes. Chu et al. (2007) have demonstrated that many standard machine learning algorithms can be phrased as MapReduce tasks, thus illuminating the versatility of this framework. In relation to language models, Brants et al. (2007) recently proposed a distributed MapReduce infrastructure to build Ngram language models having up to 300 billion n-grams. We take inspiration from this and use the MapReduce infrastructure for our DLMs. Also, the MapReduce paradigm allows us to easily fit different variants of our learning algorithm in a modular fashion by only making small changes to the MapReduce functions.

Algorithm 1 Distributed Perceptron [19]

Require: Training samples $\mathcal{T} = \{x_i, y_i\}_{i=1}^{\mathcal{N}}$
$\mathbf{w}^0 := [0, \ldots, 0]$
Partition $\mathcal{T}$ into $\mathcal{C}$ parts, $\mathcal{T}_1, \ldots, \mathcal{T}_{\mathcal{C}}$
$[\mu_1, \ldots, \mu_{\mathcal{C}}] := [\frac{1}{\mathcal{C}}, \ldots, \frac{1}{\mathcal{C}}]$
for $t := 1$ to T **do**
  for $c := 1$ to $\mathcal{C}$ **do**
    $\mathbf{w} := \mathbf{w}^{t-1}$
    for $j := 1$ to $|\mathcal{T}_c|$ **do**
      $\tilde{y}_{c,j}^t := \text{argmax}_y \, \mathbf{w} \cdot \mathbf{\Phi}(x_{c,j}, y)$
      $\boldsymbol{\delta} := \mathbf{\Phi}(x_{c,j}, y_{c,j}) - \mathbf{\Phi}(x_{c,j}, \tilde{y}_{c,j}^t)$
      $\mathbf{w} := \mathbf{w} + \boldsymbol{\delta}$
    end for
    $\mathbf{w}_c^t := \mathbf{w}$
  end for
  $\mathbf{w}^t := \sum_{c=1}^{\mathcal{C}} \mu_c \mathbf{w}_c^t$
end for
return $\mathbf{w}^T$

In the MapReduce framework, any computation is expressed as two user-defined functions: *Map* and *Reduce*. The *Map* function takes as input a key/value pair and processes it using user-defined functions to generate a set of intermediate key/value pairs. The *Reduce* function receives all intermediate pairs that are associated with the same key value.

The distributed nature of this framework comes from the ability to invoke the *Map* function on different parts of the input data simultaneously. Since the framework assures that all the values corresponding to a given key will be accummulated at the end of all the *Map* invocations on the input data, different machines can simultaneously execute the *Reduce* to operate on different parts of the intermediate data.

Any MapReduce application typically implements *Mapper*/*Reducer* interfaces to provide the desired *Map*/*Reduce* functionalities. For our models, we use two different Mappers (as illustrated in Fig. 8.6) to compute feature weights for one training epoch. The *Rerank-Mapper* receives as input a set of training utterances and has the capacity to request feature weights computed in the previous training epoch. *Rerank-Mapper* then computes feature updates for the given training data (the subset of the training data received by a single *Rerank-Mapper* instance will be henceforth referred to as a "Map chunk"). We also have a second *Identity-Mapper* that receives feature weights from the previous training epoch and directly maps the inputs to outputs which are provided to the *Reducer*. The *Reducer* combines the outputs from both *Rerank-Mapper* and *Identity-Mapper* and outputs the feature weights for the current training epoch. These output feature weights are persisted on disk in the form of SSTables that are an efficient abstraction to store large numbers of key-value pairs.

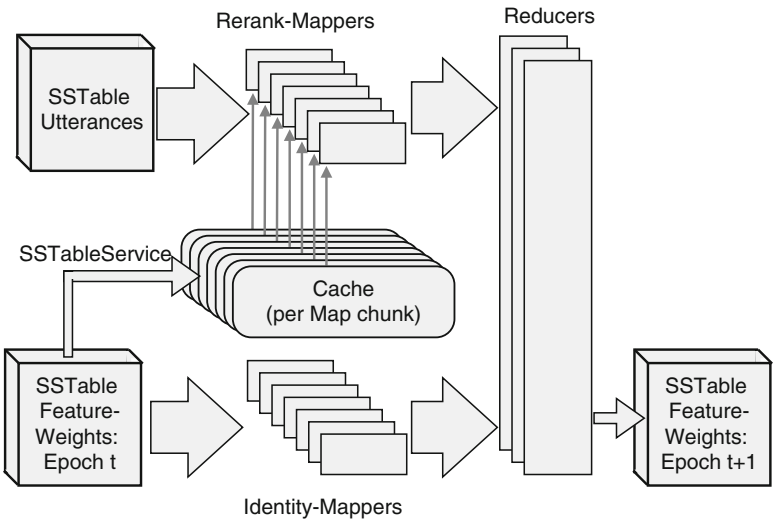

Fig. 8.6 MapReduce implementation of reranking using discriminative language models

The features corresponding to a Map chunk at the end of training epoch need to be made available to *Rerank-Mapper* in the subsequent training epoch. Instead of accessing the features on demand from the SSTables that store these feature weights, every *Rerank-Mapper* stores the features needed for the current Map chunk in a cache. Though the number of features stored in the SSTables are determined by the total number of training utterances, the number of features that are accessed by a *Rerank-Mapper* instance are only proportional to the chunk size and can be cached locally. This is an important implementation choice because it allows us to estimate very large distributed models: the bottleneck is no longer the total model size but rather the cache size that is in turn controlled by the Map chunk size. Section "Evaluating Our DLM Rescoring Framework on *weakAM-dev/test*" discusses in more detail about different model sizes and the effects of varying Map chunk size on recognition performance.

Figure 8.6 is a schematic diagram of our entire framework; Fig. 8.7 shows a more detailed representation of a single *Rerank-Mapper*, an *Identity-Mapper* and a *Reducer*, with the pseudocode of these interfaces shown inside their respective boxes. *Identity-Mapper* gets feature weights from the previous training epoch as input ($\mathbf{w}^t$) and passes them to the output unchanged. *Rerank-Mapper* calls the function `Rerank` that takes an N-best list of a training utterance (**utt.Nbest**) and the current feature weights ($\mathbf{w}_{curr}$) as input and reranks the N-best list to obtain the best scoring hypothesis. If this differs from the correct transcript for `utt`, `FeatureDiff` computes the difference in feature vectors corresponding to the two hypotheses (we call it δ) and $\mathbf{w}_{curr}$ is incremented with δ. `Emit` is the output function of a

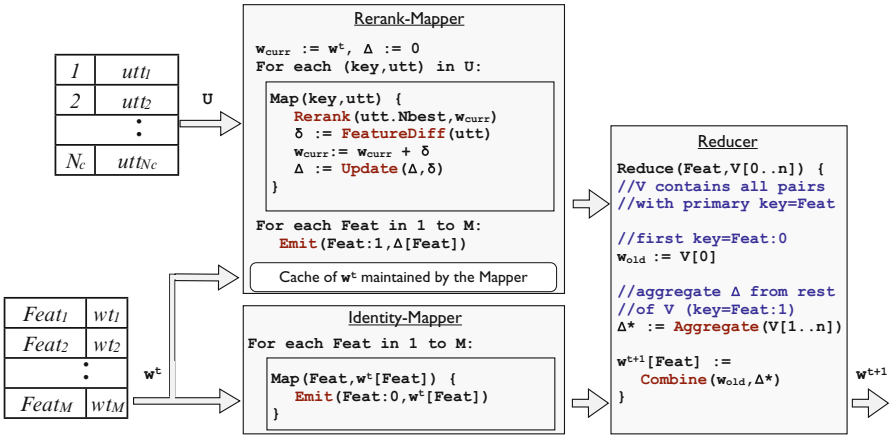

Fig. 8.7 Details of the Mapper and Reducer

Naive Distributed Perceptron:
- Update(Δ, δ) returns $\Delta + \delta$.
- Aggregate$([\Delta_1^t, \ldots, \Delta_{\mathcal{C}}^t])$ returns $\Delta^* = \sum_{c=1}^{\mathcal{C}} \Delta_c^t$.
- Combine(w_{NP}^{t-1}, Δ^*) returns $w_{NP}^{t-1} + \Delta^*$.

Distributed Perceptron:
- Update and Combine are as for the *Naive Distributed Perceptron*.
- Aggregate$([\Delta_1^t, \ldots, \Delta_{\mathcal{C}}^t])$ returns $\Delta^* = \sum_{c=1}^{\mathcal{C}} \mu_c \Delta_c^t$.

Averaged Distributed Perceptron: Here, $w^t = (w_{AV}^t, w_{DP}^t)$, and $\Delta = (\boldsymbol{\beta}, \boldsymbol{\alpha})$ contain pairs of values; $\boldsymbol{\alpha}$ is used to maintain w_{DP}^t and $\boldsymbol{\beta}$, both of which in turn are used to maintain w_{AV}^t ($\boldsymbol{\alpha}_c^t$ plays the role of Δ_c^t in *Distributed Perceptron*). Only w_{AV}^t is used in the final evaluation and only w_{DP}^t is used during training.
- Update$((\boldsymbol{\beta}, \boldsymbol{\alpha}), \delta)$ returns $(\boldsymbol{\beta} + \boldsymbol{\alpha} + \delta, \boldsymbol{\alpha} + \delta)$.
- Aggregate$([\Delta_1^t, \ldots, \Delta_{\mathcal{C}}^t])$ where $\Delta_c^t = (\boldsymbol{\beta}_c^t, \boldsymbol{\alpha}_c^t)$, returns $\Delta^* = (\boldsymbol{\beta}^*, \boldsymbol{\alpha}^*)$ where $\boldsymbol{\beta}^* = \sum_{c=1}^{\mathcal{C}} \boldsymbol{\beta}_c^t$, and $\boldsymbol{\alpha}^* = \sum_{c=1}^{\mathcal{C}} \mu_c \boldsymbol{\alpha}_c^t$.
- Combine$((w_{AV}^{t-1}, w_{DP}^{t-1}), (\boldsymbol{\beta}^*, \boldsymbol{\alpha}^*))$ returns $(\frac{t-1}{t} w_{AV}^{t-1} + \frac{1}{t} w_{DP}^{t-1} + \frac{1}{\mathcal{N}t} \boldsymbol{\beta}^*, w_{DP}^{t-1} + \boldsymbol{\alpha}^*)$.

Fig. 8.8 `Update`, `Aggregate` and `Combine` procedures for the three variants of the distributed perceptron algorithm

Mapper that outputs a processed key/value pair. For every feature `Feat`, both *Identity-Mapper* and *Rerank-Mapper* also output a secondary key (0 or 1, respectively); this is denoted as `Feat`:0 and `Feat`:1. At the *Reducer*, its inputs arrive sorted according to the secondary key; thus, the feature weight corresponding to `Feat` from the previous training epoch produced by *Identity-Mapper* will necessarily arrive before `Feat`'s current updates from the *Rerank-Mapper*. This ensures that $\mathbf{w}^{t+1}$ is updated correctly starting with $\mathbf{w}^t$. The functions `Update`, `Aggregate` and `Combine` are explained in the context of three variants of the distributed perceptron algorithm in Fig. 8.8.

MapReduce Variants of the Distributed Perceptron Algorithm

Our MapReduce setup described in the previous section allows for different variants of the distributed perceptron training algorithm to be implemented easily. We experimented with three slightly differing variants of a distributed training strategy for the structured perceptron, *Naive Distributed Perceptron*, *Distributed Perceptron* and *Averaged Distributed Perceptron*; these are defined in terms of Update, Aggregate and Combine in Fig. 8.8 where each variant can be implemented by plugging in these definitions from Fig. 8.8 into the pseudocode shown in Fig. 8.7. We briefly describe the functionalities of these three variants. The weights at the end of a training epoch t for a single feature f are $(w_{NP}{}^t, w_{DP}{}^t, w_{AV}{}^t)$ corresponding to *Naive Distributed Perceptron*, *Distributed Perceptron* and *Averaged Distributed Perceptron*, respectively; $\varphi(\cdot,\cdot)$ correspond to feature f's value in Φ from Algorithm 1. Below, $\delta^t_{c,j} = \phi(x_{c,j}, y_{c,j}) - \phi(x_{c,j}, \tilde{y}^t_{c,j})$ and $\mathcal{N}_c$ = number of utterances in Map chunk $\mathcal{T}_c$.

At the end of epoch t, the weight increments in that epoch from all map chunks are added together and added to $w_{NP}{}^{t-1}$ to obtain $w_{NP}{}^t$.

Here, instead of adding increments from the map chunks, at the end of epoch t, they are averaged together using weights μ_c, $c = 1$ to $\mathcal{C}$, and used to increment $w_{DP}{}^{t-1}$ to $w_{DP}{}^t$.

In this variant, firstly, all epochs are carried out as in the Distributed Perceptron algorithm above. But at the end of t epochs, all the weights encountered during the whole process, over all utterances and all chunks, are averaged together to obtain the final weight $w_{AV}{}^t$. Formally,

$$w^t_{AV} = \frac{1}{\mathcal{N} \cdot t} \sum_{t'=1}^{t} \sum_{c=1}^{\mathcal{C}} \sum_{j=1}^{\mathcal{N}_c} w^{t'}_{c,j},$$

where $w_{c,j}{}^t$ refers to the current weight for map chunk c, in the tth epoch after processing j utterances and $\mathcal{N}$ is the total number of utterances. In our implementation, we maintain only the weight $w_{DP}{}^{t-1}$ from the previous epoch, the cumulative increment $\gamma^t_{c,j} = \sum_{k=1}^{j} \delta^t_{c,k}$ so far in the current epoch, and a running average $w_{AV}{}^{t-1}$. Note that, for all c, j, $w^t_{c,j} = w^{t-1}_{DP} + \gamma^t_{c,j}$, and hence

$$\begin{aligned} \mathcal{N}t \cdot w^t_{AV} &= \mathcal{N}(t-1) w^{t-1}_{AV} + \sum_{c=1}^{\mathcal{C}} \sum_{j=1}^{\mathcal{N}_c} w^t_{c,j} \\ &= \mathcal{N}(t-1) w^{t-1}_{AV} + \mathcal{N} w^{t-1}_{DP} + \sum_{c=1}^{\mathcal{C}} \beta^t_c \end{aligned}$$

where $\beta^t_c = \sum_{j=1}^{\mathcal{N}} \gamma^t_{c,j}$. Writing $\beta^* = \sum_{c=1}^{\mathcal{C}} \beta^t_c$, we have $w^t_{AV} = \frac{t-1}{t} w^{t-1}_{AV} + \frac{1}{t} w^{t-1}_{DP} + \frac{1}{\mathcal{N}t} \beta^*$.

Experiments and Results

Our DLMs are evaluated in two ways: (1) we extract a development set (*weakAM-dev*) and a test set (*weakAM-test*) from the speech data that is re-decoded with a *weakAM*, and (2) we use a standard voice search test set (*v-search-test*) (Strope et al. 2011) to evaluate actual ASR performance on voice search. More details regarding our experimental setup along with a discussion of our experiments and results are described in the rest of the section.

Experimental Setup

We generate training lattices using speech data that is re-decoded with a *weakAM* acoustic model and the baseline language model. We use maximum likelihood trained single mixture Gaussians for our *weakAM*. And, we use a sufficiently small baseline LM (~21 million n-grams) to allow for sub-real time lattice generation on the training data with a small memory footprint, without compromising on its strength. Chelba et al. (2010) demonstrate that it takes much larger LMs to get a significant relative gain in WER. Our largest discriminative language models are trained on 87,000 h of speech, or ~350 million words (*weakAM-train*) obtained by filtering voice search logs at 0.8 confidence, and re-decoding the speech data with a *weakAM* to generate N-best lists. We set aside a part of this *weakAM-train* data to create *weakAM-dev* and *weakAM-test*: these data sets consist of 328,460/316,992 utterances, or 1,182,756/1,129,065 words, respectively.

We use a manually-transcribed, standard voice search test set (*v-search-test*) consisting of 27,273 utterances, or 87,360 words to evaluate actual ASR performance using our *weakAM*-trained models. All voice search data used in the experiments is anonymized.

Figure 8.9 shows oracle error rates, both at the sentence and word level, using N-best lists of utterances in *weakAM-dev* and *v-search-test*. These error rates are obtained by choosing the best of the top N hypotheses that is either an exact match (for sentence error rate) or closest in edit distance (for word error rate) to the correct transcript. The N-best lists for *weakAM-dev* are generated using a weak AM and N-best lists for *v-search-test* are generated using the baseline (strong) AM. Figure 8.9 shows these error rates plotted against a varying threshold N for the N-best lists. Note there are sufficient word errors in the *weakAM* data to train DLMs; also, we observe that the plot flattens out after N = 100, thus informing us that N = 100 is a reasonable threshold to use when training our DLMs.

Experiments in section "Evaluating Our DLM Rescoring Framework on *weakAM-dev/test*" involve evaluating our learning setup using *weakAM-dev/test*. We then investigate whether improvements on *weakAM-dev/test* translate to *v-search-test* where N-best are generated using the *strongAM*, and scored against *manual* transcripts using fully fledged text normalization instead of the string edit

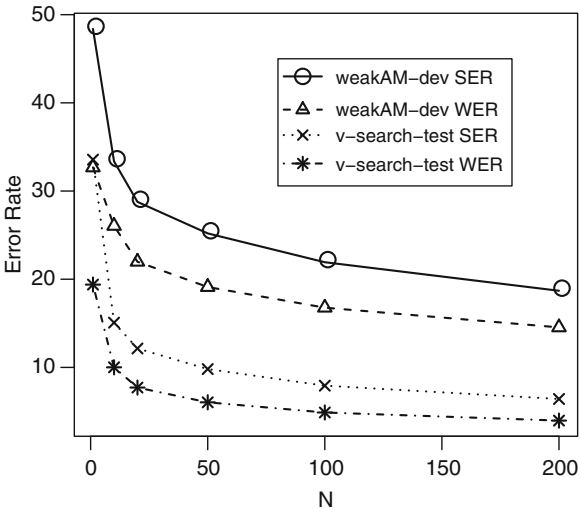

Fig. 8.9 Oracle error rates at word/sentence level for *weakAM-dev* with the weak AM and *v-search-test* with the baseline AM

distance used in training the DLM. More details about the implications of this text normalization on WER can be found in section "Evaluating ASR Performance on *v-search-test* Using DLM Rescoring".

Evaluating Our DLM Rescoring Framework on weakAM-dev/test

Improvements on *weakAM-dev* Using Different Variants of Training for theDLMs

We evaluate the performance of all the variants of the distributed perceptron algorithm described in section "MapReduce Implementation Details" over 10 training epochs using a DLM trained on ~ 20,000 h of speech with trigram word features. Figure 8.10 shows the drop in WER for all the three variants. We observe that the *Naive Distributed Perceptron* gives modest improvements in WER compared to the baseline WER of 32.5%. However, averaging over the number of Map chunks as in the *Distributed Perceptron* or over the total number of utterances and total number of training epochs as in the *Averaged Distributed Perceptron* significantly improves recognition performance; this is in line with the findings reported in Collins (2002) and McDonald et al. (2010) of averaging being an effective way of adding regularization to the perceptron algorithm.

Our best-performing *Distributed Perceptron* model gives a 4.7 % absolute (~ 15% relative) improvement over the baseline WER of 1-best hypotheses in *weakAM-dev*. This, however, could be attributed to a combination of factors: the use of large amounts of additional training data for the DLMs or the discriminative

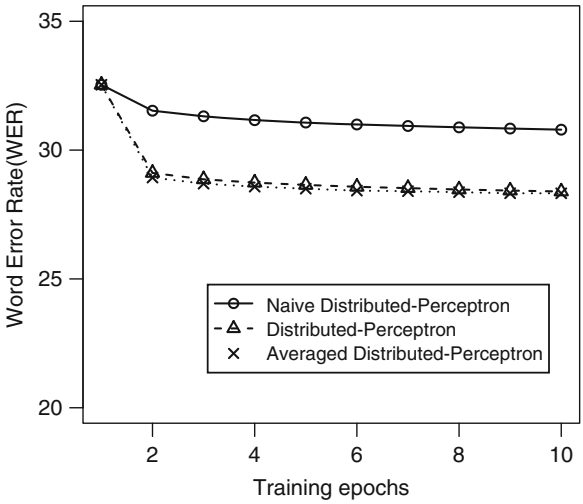

Fig. 8.10 Word error rates on *weakAM-dev* using *Perceptron*, *Distributed Perceptron* and *AveragedPerceptron* models

nature of the model. In order to isolate the improvements brought upon mainly by the second factor, we build an ML trained backoff trigram LM (ML-3 g) using the reference transcripts of all the utterances used to train the DLMs. The N-best lists in *weakAM-dev* are reranked using ML-3 g probabilities linearly interpolated with the LM probabilities from the lattices. We also experiment with a log-linear interpolation of the models; this performs slightly worse than rescoring with linear interpolation.

Impact of Varying Orders of N-gram Features

Table 8.9 shows that our best performing model (DLM-3 g) gives a significant ~2% absolute (~6% relative) improvement over ML-3 g. We also observe that most of the improvements come from the unigram and bigram features. We do not expect higher order N-gram features to significantly help recognition performance; we further confirm this by building DLM-4 g and DLM-5 g that use up to 4- and 5-g word features, respectively. Table 8.10 gives the progression of WERs for 6 epochs using DLM-3 g, DLM-4 g and DLM-5 g showing minute improvements as we increase the order of Ngram features from 3 to 5.

Impact of Model Size on WER

We experiment with varying amounts of training data to build our DLMs and assess the impact of model size on WER. These are evaluated on the test set derived from the weakAM data (*weakAM-test*). Table 8.11 shows each model along with its size

Table 8.9 WERs on *weakAM-dev* using the baseline 1-best system, ML-3 g and DLM-1/2/3 g

Data set	Baseline (%)	ML-3 g (%)	DLM-1 g (%)	DLM-2 g (%)	DLM-3 g (%)
weakAM-dev	32.5	29.8	29.5	28.3	27.8

Table 8.10 WERs on *weakAM-dev* using DLM-3 g, DLM-4 g and DLM-5 g of 6 training epochs

Iteration	DLM-3 g (%)	DLM-4 g (%)	DLM-5 g (%)
1	32.53	32.53	32.53
2	29.52	29.47	29.46
3	29.26	29.23	29.22
4	29.11	29.08	29.06
5	29.01	28.98	28.96
6	28.95	28.90	28.87

Table 8.11 WERs on *weakAM-test* using DLMs of varying sizes

Model	Size (in millions)	Coverage (%)	WER (%)
Baseline	21	–	39.08
Model 1	65	74.8	34.18
Model 2	135	76.9	33.83
Model 3	194	77.8	33.74
Model 4	253	78.4	33.68

(measured in total number of word features), coverage on *weakAM-test* in percent of tokens (number of word features in *weakAM-test* that are in the model) and WER on *weakAM-test*. As expected, coverage increases with increasing model size with a corresponding tiny drop in WER as the model size increases. "Larger models", built by increasing the number of training utterances used to train the DLMs, do not yield significant gains in accuracy. We need to find a good way of adjusting the model capacity with increasing amounts of data.

Impact of Varying Map Chunk Sizes

We also experiment with varying Map chunk size to determine its effect on WER. Figure 8.11 shows WERs on *weakAM-dev* using our best *Distributed Perceptron* model with different Map chunk sizes (64 MB, 512 MB, 2 GB). For clarity, we examine two limit cases: (a) using a single Map chunk for the entire training data is equivalent to the conventional structured perceptron and (b) using a single training instance per Map chunk is equivalent to batch training. We observe that moving from 64 MB to 512 MB significantly improves WER and the rate of improvement in WER decreases when we increase the Map chunk size further to 2 GB. We attribute

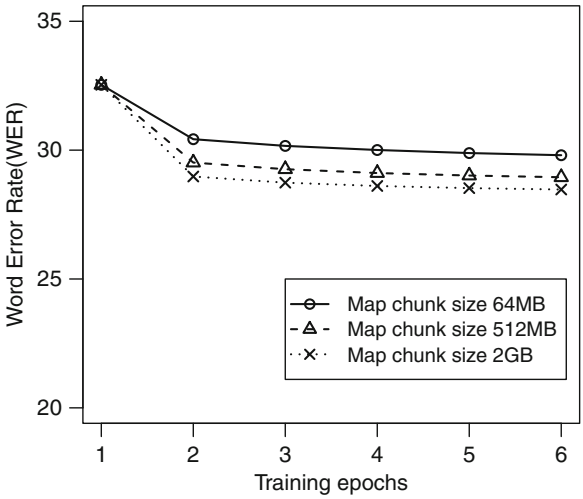

Fig. 8.11 Word error rates on *weakAM-dev* using varying Map chunk sizes of 64 MB, 512 MB and 2 GB

these reductions in WER with increasing Map chunk size to on-line parameter updates being done on increasing amounts of training samples in each Map chunk. The larger number of training samples per Map chunk accounts for greater stability in the parameters learned by each Map chunk in our DLM.

Evaluating ASR Performance on v-search-test Using DLM Rescoring

We evaluate our best *Distributed Perceptron* DLM model on *v-search-test* lattices that are generated using a strong AM. We hope that the large relative gains on *weakAM-dev/test* translate to similar gains on this standard voice search data set as well. Table 8.12 shows the WERs on both *weakAM-test* and *v-search-test* using Model 1 (from Table 8.11).[2] We observe a small but statistically significant ($p < 0.05$) reduction ($\sim 2\%$ relative) in WER on *v-search-test* over reranking with a linearly interpolated ML-3 g. This is encouraging because we attain this improvement using training lattices that were generated using a considerably weaker AM.

It is instructive to analyze why the relative gains in performance on *weakAM-dev/test* do not translate to *v-search-test*. Our DLMs are built using N-best outputs from the recognizer that live in the "spoken domain" (SD) and the manually transcribed *v-search-data* transcripts live in the "written domain" (WD). The normalization of training data from WD to SD is as described in Chelba et al. (2010);

[2] We also experimented with the larger Model 4 and saw similar improvements on *v-search-test* as with Model 1.

Table 8.12 WERs on *weakAM-test* and *v-search-test*

Data set	Baseline (%)	ML-3 g (%)	DLM-3 g (%)
weakAM-test	39.1	36.7	34.2
v-search-test	14.9	14.6	14.3

inverse text normalization (ITN) undoes most of that when moving text from SD to WD, and it is done in a heuristic way. There is ~2% absolute reduction in WER when we move the N-best from SD to WD via ITN; this is how WER on *v-search-test* is computed by the voice search evaluation code. Contrary to this, in DLM training we compute WERs using string edit distance between test data transcripts and the N-best hypotheses and thus we ignore the mismatch between domains WD and SD. It is quite likely that part of what the DLM learns is to pick N-best hypotheses that come closer to WD, but may not truly result in WER gains after ITN. This would explain part of the mismatch between the large relative gains on *weakAM-dev/test* compared to the smaller gains on *v-search-test*. We could correct for this by applying ITN to the N-best lists from SD to move to WD before computing the oracle best in the list. An even more desirable solution is to build the LM directly on WD text; text normalization would be employed for pronunciation generation, but ITN is not needed anymore (the LM picks the most likely WD word string for homophone queries at recognition).

Optimal Size, Freshness and Time-Frame for Voice Search Vocabulary

In this section, we investigate how to optimize the vocabulary for a voice search language model. The metric we optimize over is the out-of-vocabulary (OOV) rate since it is a strong indicator of user experience; the higher the OOV rate, the more likely the user is to have a poor experience. Clearly, each OOV word will result in at least one error at the word level,[3] and in exactly one error at the whole query/sentencelevel. In ASR practice, OOV rates below 0.01 (1%) are deemed acceptable since typical WER values are well above 10%.

As shown in Chelba et al. (2010), a typical vocabulary for a US English voice search language model (LM) is trained on the US English query stream, contains about 1 million words, and achieves out-of-vocabulary (OOV) rate of 0.57% on unseen text query data, after query normalization.

In a departure from typical vocabulary estimation methodology, (Jelinek 1990; Venkataraman and Wang 2003), the web search query stream not only provides us

[3]The approximate rule of thumb is 1.5 errors for every OOV word, so an OOV rate of 1% would lead to about 1.5% absolute loss in word error rate (WER).

with training data for the LM, but also with session level information based on 24-h cookies. Assuming that each cookie corresponds to the experience of a web search user over exactly 1 day, we can compute per-one-day-user OOV rates, and directly corelate them with the voice search LM vocabulary size (Kamvar and Chelba 2012).

Since the vocabulary estimation algorithms are extremely simple, the work presented here is purely experimental. Our methodology is as follows:

- Select as training data $\mathcal{T}$ a set of queries arriving at the `google.com` front-end during time period T;
- Text normalize the training data, see section "A Note on Query Normalization";
- Estimate a vocabulary $\mathcal{V}$ by thresholding the 1-g count of words in $\mathcal{T}$ such that it exceeds C, $\mathcal{V}(T,C)$;
- Select as test data $\mathcal{T}$ a set of queries arriving at the google.com front-end during time period E; E is a single day that occurs after T, and the data is subjected to the exact same text normalization used in training;
- We evaluate both *aggregate* and *per-cookie* OOV rates, and report the aggregate OOV rate across all words in $\mathcal{T}$, as well as the percentage of cookies in $\mathcal{T}$ that experience an OOV rate that is less or equal than 0.01 (1%).

We aim to answer the following questions:

- How does the vocabulary size (controlled by the threshold C) impact both *aggregate* and *per-cookie* OOV rates?
- How does the vocabulary freshness (gap between T and E) impact the OOV rate?
- How does the time-frame (duration of T) of the training data $\mathcal{T}$ used to estimate the vocabulary $\mathcal{V}(T,C)$ impact the OOV rate?

A Note on Query Normalization

We build the vocabulary by considering all US English queries logged during T. We break each query up into words, and discard words that have non-alphabetic characters. We perform the same normalization for the test set. So for example if the queries in $\mathcal{T}$ were: `gawker.com, pizza san francisco, baby food,4chan status` the resulting vocabulary would be `pizza, san, francisco,baby, food, status`. The query `gawker.com` and the word `4chan` would not be included in the vocabulary because they contain non-alphabetic characters.

We note that the above query normalization is extremely conservative in the sense that it discards a lot of problematic cases, and keeps the vocabulary sizes and OOV rates smaller than what would be required for building a vocabulary and language model that would actually be used for voice search query transcription. As a

Table 8.13 Vocabulary size as a function of count threshold

Threshold	Vocabulary size
15	3,643,583
30	2,277,696
60	1,429,888
120	901,213
240	569,330
480	361,776
960	232,808

result, the vocabulary sizes that we report to achieve certain OOV values are very likely just lower bounds on the actual vocabulary sizes needed, were correct text normalization (see Chelba et al. 2010 for an example text normalization pipeline) to be performed.

Experiments

The various vocabularies used in our experiment are created from queries issued during a 1-week–1-month period starting on 10/04/2011. The vocabulary is comprised of the words that were repeated C or more times in $\mathcal{T}$. We chose seven values for C: 960, 480, 240, 120, 60, 30 and 15. As C decreases, the vocabulary size increases; to preserve user privacy we do not use C values lower than 15. For each training set $\mathcal{T}$ discussed in this paper, we will create seven different vocabularies based on these thresholds.

Each test set $\mathcal{T}$ is comprised of queries associated with a set of over 10 million cookies during a 1-day period. We associate test queries by cookie-id in order to compute user-based (per-cookie) OOV rate.

All of our data is strictly anonymous; the queries bear no user-identifying information. The only query data saved after training are the vocabularies. The evaluation on test data is done by counting on streamed filtered query logs, without saving any data.

Vocabulary Size

To understand the impact of vocabulary size on OOV rate, we created several vocabularies from the queries issued in the week $T = 10/4/2011 - 10/10/2011$. The size of the various vocabularies as a function of the count threshold is presented in Table 8.13; Fig. 8.12 shows the relationship between the logarithm of the size of the vocabulary and the aggregate OOV rate—a log-log plot of the same data points

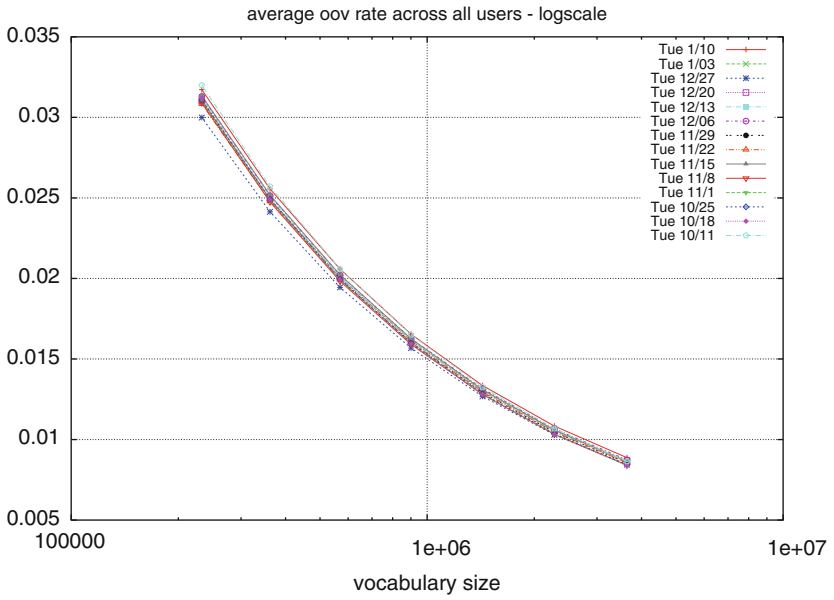

Fig. 8.12 Aggregate OOV rate as a function of vocabulary size (log-scale), evaluated on a range of test sets collected every Tuesday between 2011/10/11 and 2012/01/03

would reveal a "quasi-linear" dependency. We have also measured the percentage of cookies/users for a given OOV rate (0.01, or 1%), and the results are shown in Fig. 8.13. At a vocabulary size of 2.25 million words ($C = 30$, aggregate OOV = 0.01), over 90% of users will experience an OOV rate of 0.01.

Vocabulary Freshness

To understand the impact of the vocabulary freshness on the OOV rate, we take the seven vocabularies described above ($T = 10/4/2011 - 10/10/2011$ and $C = 960, 480, 240, 120, 60, 30, 15$) and investigate the OOV rate change as the lag between the training data T and the test data E increases: we used the 14 consecutive Tuesdays between 2010/10/11 and 2011/01/20 as test data. We chose to keep the day of week consistent (a Tuesday) across this set of E dates in order to mitigate any confounding factors with regard to day-of-week.

We found that within a 14-week time span, as the *freshness* of the vocabulary decreases, there is no consistent increase in the aggregate OOV rate (Fig. 8.12) nor any significant decrease in the percentage of users who experience less than 0.01 (1%) OOV rate (Fig. 8.13).

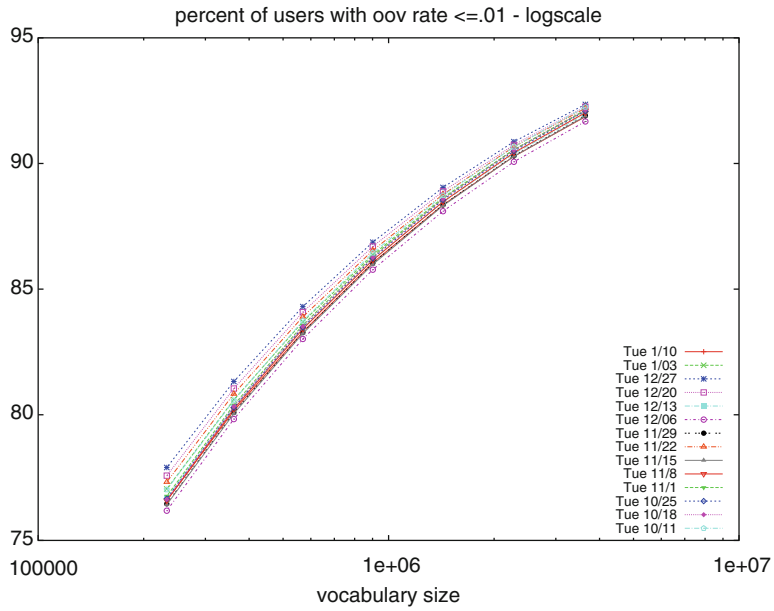

Fig. 8.13 Percentage of cookies/users with OOV rate less than 0.01 (1%) as a function of vocabulary size (log-scale), evaluated on test sets collected every Tuesday between 2011/10/11 and 2012/01/03

Vocabulary Time Frame

To understand how the duration of T (the time window over which the vocabulary is estimated) impacts OOV rate, we created vocabularies over the following time windows:

- 1 week period between 10/25/2011 and 10/31/2011
- 2 week period between 10/18/2011 and 10/31/2011
- 3 week period between 10/11/2011 and 10/31/2011
- 4 week period between 10/04/2011 and 10/31/2011

We again created seven threshold based vocabularies for each T. We evaluate the aggregate OOV rate on the date $E = 11/1/2011$, see Fig. 8.14, as well as the percentage of users with a per-cookie OOV rate below 0.01 (1%), see Fig. 8.15. We see that the shape of the graph is fairly consistent across T time windows, and a week of training data is as good as a month.

More interestingly, Fig. 8.15 shows that aiming at an operating point where 95% the percentage of users experience OOV rates below 0.01 (1%) requires significantly larger vocabularies, approx. 10 million words.

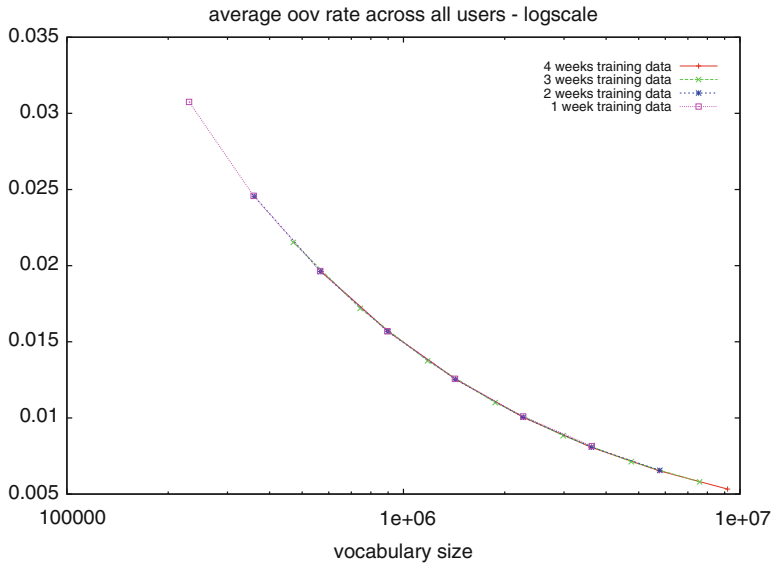

Fig. 8.14 Aggregate OOV rate on 11/1/2011 over vocabularies built from increasingly large training sets

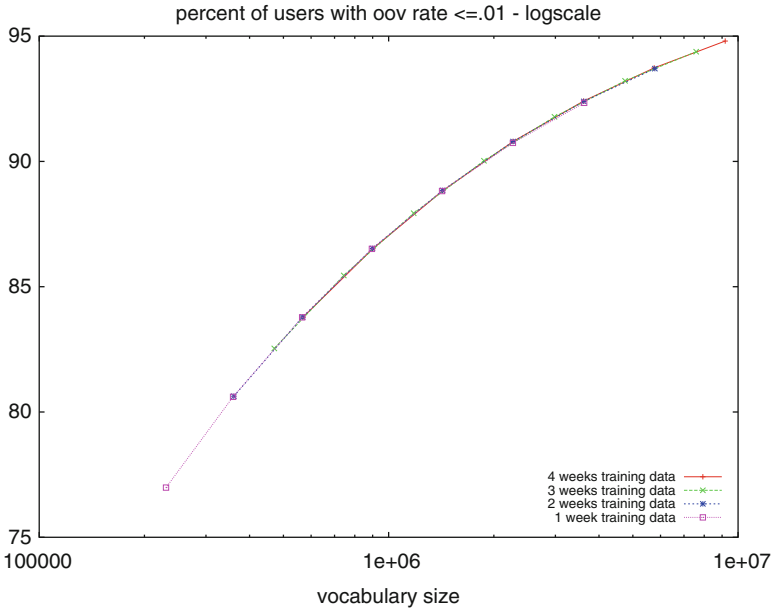

Fig. 8.15 Percentage of cookies/users with OOV rate less than 0.01 (1%) on 11/1/2011 over vocabularies built from increasingly large training sets

Conclusions: Language Modeling for Voice Search

Our experiments show that with careful text normalization the query stream is not as "wild" as it seems at first sight. One can achieve excellent OOV rates for a 1 million word vocabulary, and n-gram hit ratios of 77/88% even at $n = 5 / 4$, respectively.

Experimental evidence suggests that the query stream is non-stationary, and that more data does not automatically imply better models even when the data is clearly matched to the test data. More careful experiments are needed to adjust model capacity and identify an optimal way of blending older and recent data—attempting to separate the stationary/non-stationary components in the query stream. Less surprisingly, we have shown that locale matters significantly for English query data across USA, Great Britain and Australia.

We generally see excellent correlation of WER with PPL under various pruning regimes, as long as the training set and vocabulary stays constant.

As for leveraging the speech logs data for better language modeling, we successfully build large-scale discriminative N-gram language models with lattices regenerated using a weak AM and derive small but significant gains in recognition performance on a voice search task where the lattices are generated using a stronger AM. We use a very simple weak AM and this suggests that there is room for improvement if we use a slightly better "weak AM". Also, we have a scalable and efficient MapReduce implementation that is amenable to adapting minor changes to the training algorithm easily and allows for training large LMs. The latter functionality will be particularly useful if we generate the contrastive set by sampling from text instead of re-decoding logs (Jyothi and Fosler-Lussier 2010).

A more careful analysis of vocabulary estimation for voice search shows that a significantly larger vocabulary (approx. 10 million words) seems to be required to guarantee a 0.01 (1%) OOV rate for 95% of the users.

Studies on the www pages side (Brants) show that after just a few million words, vocabulary growth is close to a straight line in the logarithmic scale; the vocabulary grows by about 69% each time the size of the text is doubled even when using one trillion words of training data. Since queries are used for finding such pages, the growth in query stream vocabulary size is easier to understand.

We also find that 1 week is as good as 1 month of data for estimating the vocabulary, and that there is very little drift in OOV rate as the test data (1 day) shifts during the 3 months following the training data used for estimating the vocabulary.

References

Allauzen C, Riley M, Schalkwyk J, Skut W, Mohri M. (2007) OpenFst: a general and efficient weighted finite-state transducer library. In: Proceedings of the ninth international conference on implementation and application of automata, (CIAA 2007). Lecture notes in computer science, vol 4783. Springer, pp 11–23. http://www.openfst.org

Allauzen C, Schalkwyk J, Riley M (2009) A generalized composition algorithm for weighted finite-state transducers. In: Proceedings of Interspeech, Brighton, pp 1203–1206

Banko M, Brill E (2001) Mitigating the paucity-of-data problem: exploring the effect of training corpus size on classifier performance for natural language processing. In: Proceedings of the first international conference on human language technology research, HLT '01, San Diego. Association for Computational Linguistics, Stroudsburg, pp 1–5

Brants T Vocabulary growth. In: Kordoni V (ed) Festschrift for Hans Uszkoreit. CSLI Publications, to appear, Stanford, CA 94305

Brants T, Xu P (2009) Distributed language models. In: HLT-NAACL tutorial abstracts, Boulder pp 3–4

Brants T, Popat AC, Xu P, Och FJ, Dean J (2007) Large language models in machine translation. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Prague, pp 858–867

Chelba C, Schalkwyk J, Brants T, Ha V, Harb B, Neveitt W, Parada C, Xu P (2010) Query language modeling for voice search In: Proceedings of SLT, Berkeley

Chu CT, Kim SK, Lin YA, Yu YY, Bradski G, Ng AY, Olukotun K (2007) Map-reduce for machine learning on multicore. Proc NIPS 19:281

Collins M (2002) Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In: Proceedings of EMNLP, Philadelphia

Gao J, Yu H, Yuan W, Xu P (2005) Minimum sample risk methods for language modeling. In: Proceedings of EMNLP, Vancouver

Ghemawat S, Dean J (2004) Mapreduce: Simplified data processing on large clusters. In: Proceedings of OSDI, San Francisco

Goodman J (2001) A bit of progress in language modeling, extended version. Technical Report, Microsoft Research

Halevy A, Norvig P, Pereira F (2009) The unreasonable effectiveness of data. IEEE Intell Syst 24(2):8–12

Hall KB, Gilpin S, Mann G (2010) MapReduce/Bigtable for distributed optimization. In: NIPS LCCC Workshop, Whistler, BC

Harb B, Chelba C, Dean J, Ghemawat S (2009) Back-off language model compression. In: Proceedings of Interspeech, Brighton. ISCA, pp 325–355

Jelinek F (1990) Self-organized language modeling for speech recognition. In: Waibel A, Lee K-F (eds) Readings in speech recognition. Morgan Kaufmann Publishers, San Mateo, pp 450–506

Jelinek F (1997) Information extraction from speech and text. MIT Press, Cambridge, MA, pp 141–142. Chap. 8

Jyothi P, Fosler-Lussier E (2010) Discriminative language modeling using simulated ASR errors. In: Proceedings of Interspeech, Makuhari

Jyothi P, Johnson L, Chelba C, Strope B (2012) Distributed discriminative language models for Google voice-search. In: Proceedings of ICASSP, Kyoto

Kamvar M, Chelba C (2012) Optimal size, freshness and time-frame for voice search vocabulary. Google Tech Report

Katz S (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans Acoust Speech Signal Process 35:400–401

Kneser R, Ney H (1995) Improved backing-off for m-gram language modeling. Proc IEEE Int Conf Acoust Speech Signal Process 1:181–184

Mann G, McDonald R, Mohri M, Silberman N, Walker D (2009) Efficient large-scale distributed training of conditional maximum entropy models. In: Proceedings of NIPS, Vancouver

McDonald R, Hall K, Mann G (2010) Distributed training strategies for the structured perceptron. In: Proceedings of NAACL, Los Angeles

Och FJ (2005) Statistical machine translation: foundations and recent advances. In: Presentation at MT-Summit. Phobet, Thailand

Paul DB, Baker JM (1992) The design for the Wall Street Journal-based CSR corpus. In: Proceedings of the workshop on speech and natural language, HLT '91, Harriman, New York. Association for Computational Linguistics, Stroudsburg, pp 357–362

Roark B, Saraçlar M, Collins M, Johnson M (2007) Discriminative n-gram language modeling. Computer Speech and Language, 21(2): 373–392

Schlüter R, Müller B, Wessel F, Ney H (1999) Interdependence of language models and discriminative training. In: Proceedings of ASRU, Keystone

Strope B, Beeferman D, Gruenstein A, Lei X (2011) Unsupervised testing strategies for ASR. In: Proceedings of Interspeech, Florence

Venkataraman A, Wang W (2003) Techniques for effective vocabulary selection. Arxiv preprint cs/0306022

Zhou Z, Gao J, Soong FK, Meng H (2006) A comparative study of discriminative methods for reranking LVCSR N-best hypotheses in domain adaptation and generalization. In Proceedings of ICASSP, Toulouse

Chapter 9
Information Extraction: Robust Mention Detection Systems

Imed Zitouni, John F. Pitrelli, and Radu Florian

Abstract Information-extraction (IE) research typically focuses on clean-text inputs. However, as we will see in this chapter, an IE engine serving real-world applications yields a high rate of false alarms, due to noisy, less-well-formed input. For example, an application processing output from a multilingual media monitoring system (e.g., TV broadcast) will have to deal with noisy input as well as inaccurate automatic transcription and translation. The resulting presence of non-target-language text in this case, and non-language material interspersed in data from other applications, raise the research problem of making IE robust to such noisy input text. This chapter addresses an important IE task: improving robustness to noise for mention detection (MD). We describe the augmentation of an existing statistical MD system to reduce false alarms in the spurious passages while maintaining performance on clean input, and even improving recall. The diverse nature of input noise leads us to pursue a multi-faceted approach to robustness. We describe a multi-stage approach to robustness, reflecting the diverse nature of input noise. Detection-error-trade-off analysis is used to evaluate a MD system. In one experiment, with English as the target language, we find that on inputs from other Latin-alphabet languages, we can eliminate 97–98% of false alarms compared to an English-only baseline system, at various fixed miss rates. In another experiment, modeling situations in which genre-specific training is infeasible, we process real data drawn from a financial-transactions text containing mixed languages and data-set codes. On these data, because annotations for data sets like this are typically not available for training a mention detector, we did not include any portion of this

I. Zitouni, Ph.D. (✉)
Principal Researcher, IBM T.J. Watson Research Center, 10 Birchwood Road,
White Plains, NY 10605, USA
e-mail: izitouni@microsoft.com

J.F. Pitrelli, Ph.D. • R. Florian, Ph.D.
Researcher, IBM T.J. Watson Research Center, 1101 Kitchwan Road,
Yorktown Heights, NY 10598, USA
e-mail: pitrelli@us.ibm.com; raduf@us.ibm.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_9,
© Springer Science+Business Media New York 2013

data set in the training of the system, yet still can eliminate 60% of the false alarms at various miss rates, compared to the baseline system. These gains come with virtually no loss in accuracy on clean English text.

Introduction

Information extraction (IE) is the task of identifying, extracting, and categorizing useful information from natural language. IE is applicable both to speech and to text; however, the approach typically used for IE from speech is to apply speech recognition to convert the speech to text, and then approach the IE step itself as a text-processing task. Thus, IE research is typically approached as a text-processing problem; however, most such work focuses on clean-text input in a pre-determined language.

The scope of an information-extraction task can be arbitrarily broad, and sometimes may even require world knowledge. Lately, IE has improved to the point of being usable for some real-world tasks whose accuracy requirements are reachable with current technology. These uses include media monitoring, topic alerts, summarization, population of databases for advanced search, etc. These uses often combine IE with technologies such as machine translation, topic clustering, and information retrieval, in addition to multilingual speech recognition.

The propagation of IE technology from isolated use to aggregates with such other technologies, from NLP experts to other types of computer scientists, and from researchers to users, feeds back to the IE research community the need for additional investigation which we loosely refer to as "information-extraction robustness" research. For example:

1. Broadcast monitoring demands that IE handle as input not only clean text, but also the transcripts output by speech recognizers.
2. Multilingual applications, and the imperfection of translation technology, require IE to contend with non-target-language text input (Pitrelli et al. 2008).
3. Naïve users at times input to IE other material which deviates from clean text, such as a PDF file that "looks" like plain text.
4. Search applications require IE to deal with databases which not only contain clean text but at times exhibit other complications like mark-up codes particular to narrow, application-specific data-format standards, for example, the excerpt from a financial-transactions data set shown in Fig. 9.1.

   Legacy industry-specific standards, such as illustrated in this example, are part of long-established processes which are cumbersome to convert to a more-modern database format. Transaction data sets typically build up over a period of years, and as seen here, can exhibit peculiar mark-up interspersed with meaningful text. They also suffer complications arising from limited-size entry fields and a diversity of data-entry personnel, leading to effects like haphazard abbreviation and improper spacing, as shown. These issues greatly complicate the IE problem, particularly considering that adapting IE to such formats is hampered by the existence of a multitude of such "standards" and by lack of sufficient annotated data in each one.

```
:54D://121000358
BANK OF BOSTON
:55D:/0148280005
NEVADA DEPT.OF VET.94C RECOV.FD
-5:MAC:E19DECA8CHK:641EB09B8968

USING OF FIELD 59: ONLY /INS/ WHEN
FOLLOWED BY BCC CODE IN CASE
OF QUESTIONS DONT HESITATE TO
CONTACT US QUOTING REFERENCE
NON-STC CHARGES OR VIA E-MAIL:
YOVANKA(UL)BRATASOVA(AT)BOA.CZ.
BEST REGARDS
BANKA OBCHODNIKA, A.S. PRAGUE, CZ

:58E::ADTX//++ ADDITIONAL
INFORMATION ++ PLEASE BE
INFORMED THAT AS A RESULT OF
THE PURCHASE OFFER ENDED ON 23
MAR 2008 CALDRADE LTD. IS
POSSESSING WITH MORE THEN 90
PER CENT VOTING RIGHT OF SLICE.
THEREFOR CALDRADE LTD. IS
EXERCISING PURCHASE RIGHTS
FOR ALL SLICE SHARES WHICH ARE
CURRENTLY NOT INHIS OWN.
PURCHASE PRICE: HUF 1.940 PER
SHARE. PLEASE :58E::ADTX//NOTE
THAT THOSE SHARES WHICH WILL
NOT BE PRESENTED TO THE OFFER
WILL BE CANCELLED AND INVALID.

:58:SIE SELBST
TRN/REF:515220 035
:78:RUECKGABE DES BETRAGES LT.
ANZBA43 M ZWECKS RUECKGABE IN
AUD. URSPR. ZU UNSEREM ZA MIT
REF. 0170252313279065 UND IHRE
RUECKG. :42:/BNF/UNSERE REF:
```

Fig. 9.1 Example application-specific text, in this case from financial transactions

A typical state-of-the-art statistical IE engine will happily process such "noisy" inputs, and will typically provide garbage-in/garbage-out performance, embarrassingly reporting spurious "information" no human would ever mistake. Yet it is also inappropriate to discard such material wholesale: even poor-quality inputs may have relevant information interspersed. This information can include accurate speech-recognition output, names which are recognizable even in wrong-language material, and clean target-language passages interleaved with the mark-up. Thus, here we address methods to make IE robust to such varied-quality inputs, and to take an integrated approach to addressing the various robustness issues tracing to

material which originates as speech as well as that originating as text. Specifically, our overall goals are

- To skip processing non-language material such as standard or database-specific mark-up,
- To process all non-target-language material cautiously, catching interspersed target-language content as well as words and phrases which are compatible with the target language, e.g. person names which are the same in the target- and non-target language, and
- To degrade gracefully when processing anomalous target-language material,

while minimizing any disruption of the processing of clean, target-language content, and avoiding any necessity for explicit pre-classification of the genre of material being input to the system. Such explicit classification would be impractical in the presence of the interleaving and the unconstrained data formats from unpredetermined sources.

To make the IE problem tractable, we focus our robustness work on one important and basic IE sub-task, mention detection (MD). MD is the task of identifying and classifying textual references to entities in open-domain texts. The *mention detection* (MD) task consists of detecting the boundary of a mention, and optionally identifying a physical object (e.g., person or organization) and other attributes. Mentions may be of type "named" (e.g. `John`, `Las Vegas`), "nominal" (e.g. `engineer`, `dentist`) or "pronominal" (e.g. `they`, `he`). A mention also has a specific class which describes the type of entity it refers to. For instance, consider the following sentence:

Chairman Steve Jobs of *Apple* said that *he* has no comments.

Here we see three mentions of one person entity: *Chairman*, *Steve Jobs* and *he*; these mentions are of type nominal, named and pronominal, respectively. We also find the named mention *Apple* referring to an organization entity. Taken in isolation, *Apple* could also be a location, as in "The Big Apple is a nickname for New York City." Like many other problems in natural language processing, such ambiguities are the major difficulties for mention detection. Also, the MD task is a more general and complex task than named-entity recognition, which aims at identifying and classifying only named mentions.

The most successful approach for mention detection has been a data-driven, statistical one. Using this approach, a set of training data is annotated by human and statistical models are learned automatically from the data. The learned model can then be applied to unseen material. Compared with a rule-based system, the statistical approach enjoys many benefits:

- A data-driven approach makes it possible to test different algorithms and features rapidly;
- A statistical system can be continually improved when new data becomes available by adding the new data to the training set;
- A statistical system can be easily ported to other languages.

Approaches discussed in this chapter are structured to separate the core MD algorithms from idiosyncrasies specific to a language. In fact, the algorithms to be presented have been used to build systems for multiple languages without major modification. This is not to say that we should ignore the language issue. Instead, language-dependent phenomena are handled by either a preprocessing step, or a configurable module that extracts features from data. For example, for highly-inflected languages such Arabic, space-delimited words may not be a good unit for MD, and a *morph* is often chosen to counter the data-sparseness problem. For languages in which valid text, including speech-recognizer output, is permitted to lack inter-word spaces, such as Chinese, Korean or Japanese, it is necessary to segment the text input to IE into words. Another example is Chinese "acronyms": new words in Chinese can be formed by concatenating either first, last, or sometimes mixed characters of multiple contiguous words. Computationally, the phenomenon can be captured by expanding the definition of acronyms to include these cases.

Our approach to IE has been to use language-independent algorithms, in order to facilitate reuse across languages, but we train them with language-specific data, for the sake of accuracy. Therefore, input is expected to be predominantly in a target language. However, real-world data genres inevitably include some mixed-language and/or non-language input. Genre-specific training is typically infeasible due to such application-specific data sets being unannotated, motivating this line of research. Therefore, the goal of this study is to investigate schemes to make a language-specific MD engine robust to the types of interspersed non-target material described above. In these initial experiments, we work with English as the target language, though we aim to make our approach to robustness as target-language-independent as possible. While our ultimate goal is a language-independent approach to robustness, in these initial experiments, English is the target language. However, we process mixed-language material including real-world data with its own peculiar mark-up, text conventions including abbreviations, and mix of languages, with the goal of English MD.

We approach robust MD using a multi-stage strategy. First, non-target-character-set passages (here, non-Latin-alphabet) are identified and marked for non-processing. Then, following word-tokenization, we apply a language classifier to a sliding variable-length set of windows in order to generate features for each word indicative of how much the text around that word resembles good English in comparison to other Latin-alphabet languages. These features are used in a separate maximum-entropy classifier whose output is a single feature to add to the MD classifier. Additional features, primarily to distinguish English from non-language input, are added to MD as well. An example is the minimum of the number of letters and the number of digits in the "word", which when greater than zero often indicates database detritus. Then we run the MD classifier enhanced with these new robustness-oriented features. We evaluate using a detection-error-trade-off (DET) (Martin et al. 1997) analysis, in addition to traditional precision/recall/*F*-measure.

This paper is organized as follows. Section "Previous Work on Mention Detection" discusses previous work. Section "Mention-Detection Algorithm" describes the baseline maximum-entropy-based MD system. Section

"Enhancements for Robustness" introduces enhancements to the system to achieve robustness. Section "Data Sets" describes databases used for experiments, which are discussed in sections "Experiment" and "Summary" draws conclusions and plots future work.

Previous Work on Mention Detection

The MD task has close ties to named-entity recognition, which has been the focus of much recent research (Benajiba et al. 2009; Bikel et al. 1997; Borthwick et al. 1998; Florian et al. 2003; Tjong Kim Sang 2002), and has been at the center of several evaluations: MUC-6, MUC-7, CoNLL'02 and CoNLL'03 shared tasks. Usually, in computational-linguistics literature, a named entity represents an instance of either a location, a person, an organization, and the named-entity-recognition task consists of identifying each individual occurrence of names of such an entity appearing in the text. In MUC-6, for example, the set of named entity types consisted of person, organization, location, time, percent and money. As stated earlier, in this paper we are interested in identification and classification of textual references to object/abstraction *mentions*, which can be either named (e.g. "Steve Jobs"), nominal (e.g. "the president") or pronominal (e.g. "she", "it"). This task has been a focus of interest in ACE since 2003. The recent ACE evaluation campaign was in 2008.

During the CoNLL'03 shared task, the system with the best performance is described in Florian et al. (2003). This system uses a linear interpolation of three different classifiers: (i) Hidden Markov Models, (ii) Maximum Entropy (ME), and (iii) Robust Risk Minimization (RRM). Their final results were 88.76 F for English and 72.41 F for German, the best results in both languages. The system described in Chieu and Tou Ng (2003) was ranked second in CoNLL'03. It is a fully maximum entropy–based approach that uses different types of features, namely, contextual and lexical features as well as capitalization. Another approach for name entity recognition in CoNLL'03 is Nguyen et al. (2003), which uses a character-based HMM approach. This method relies heavily on internal evidence for the name entities. Tri Tran et al. (2007) show that using a Support Vector Machine (SVM) approach outperforms a CRF-based model ($F_{\beta=1}$=87.75 vs. 86.48) on the name entity recognition task in Vietnamese. The comparison is based on the average F-measure obtained by using the same feature-set with both SVMs and CRFs. Benajiba and Rosso (2008) and Benajiba et al. (2009) show how a CRF-based technique is effective for mention detection and name entity recognition. They report results for Arabic on different feature sets including contextual, morphological, and lexical features, together with gazetteer-based features. There are also several papers that present results on ACE data. As an example, Florian et al. (2006) show a two-step approach, boundary detection and then classification, for mention detection. This technique leads to better perfomance when compared to a model that jointly predicts the boundary and the mention type.

Effort to handle noisy data is still limited, especially for scenarios in which the system at decoding time does not have prior knowledge of the input data source.

Previous work dealing with unstructured data assumes the knowledge of the input data source. As an example, Minkov et al. (2005) assume that the input data is text from e-mails, and define special features to enhance the detection of named entities. Miller et al. (2000) assume that the input data is the output of a speech or optical character recognition system, and hence extract new features for better named-entity recognition. In a different research problem, Yi et al. eliminate the noisy text from the document before performing data mining (Yi et al. 2003). Hence, they do not try to process noisy data; instead, they remove it. The approach we propose in this paper does not assume prior knowledge of the data source. Also we do not want to eliminate the noisy data, but rather attempt to detect the appropriate mentions, if any, that appear in that portion of the data.

Mention-Detection Algorithm

Similarly to classical NLP tasks such as base phrase chunking (Ramshaw and Marcus 1999) and named-entity recognition (Tjong Kim Sang 2002), we formulate the MD task as a sequence-classification problem, by assigning to each word token in the text a label indicating whether it starts a specific mention, is inside a specific mention, or is outside any mentions. We also assign to every non-outside label a class to specify entity type e.g. person, organization, location, etc. We are interested in a statistical approach that can easily be adapted for several languages and that has the ability to integrate easily and make effective use of diverse sources of information to achieve high system performance. This is because, similar to many NLP tasks, good performance has been shown to depend heavily on integrating many sources of information (Florian et al. 2004). In fact, we believe that the feature set used for classification has a much larger impact on the performance of the resulting system than the classification method itself.

For this chapter we investigate the Maximum Entropy Markov Model as described previously in (Florian et al. 2004; Zitouni and Florian 2009). This widely used model type is called *maximum entropy*, or *MaxEnt*. It integrates arbitrary types of information and make a classification decision by aggregating all information available for a given classification, but the reader can replace it with his/her favorite feature-based classifier throughout the chapter. Example of classifier includes Support Vector Machine (SVM), Conditional Random Field (CRF), etc.

The maximum-entropy model used in this chapted is trained using the *sequential conditional generalized iterative scaling* (SCGIS) technique (Goodman 2002), and it uses a *Gaussian prior* for regularization (Chen and Rosenfeld 2000); the resulting model cannot really be called a maximum-entropy model, as it does not yield the model which has the maximum entropy (the second term in the product), but rather is a maximum-a-posteriori model. For historical reasons, the natural language processing community often uses this name, but other communities use names such as *log-linear models* and *exponential models* to refer to this type of model.

A Brief Overview of Maximum Entropy Models

A maximum entropy model takes an example to be classified into one of a finite set of classes, projects that example into a *feature space* and then yields a probability distribution over the possible classes. Let's make matters more precise with some notation: let $\mathcal{Y} = \{y_1, \ldots, y_n\}$ be the set of predicted classes, $\mathcal{X}$ be the example space and $\mathcal{F} = \{0,1\}^m$ be a feature space. Each example $(x, y) \in \mathcal{X}$ has associated a vector of m binary features $f(x,y) = (f_1(x,y), \ldots, f_m(x,y))$.[1] The goal of the training process is to associate examples $x \in \mathcal{X}$ with either a probability distribution over the labels from $\mathcal{Y}$, $p(\cdot \mid x)$ (if we are interested in *soft* classification) or associate one label $y \in \mathcal{Y}$ (if we are interested in *hard* classification). Soft classification means that a likelihood will be attributed for every label, whereas a hard classification stands for predicting the most likely label in the current context.

A MaxEnt model associates a set of weights $\{\alpha_j\}_{j=1\ldots m}$ with the features (f_j); the higher the absolute value, the heavier the impact a particular feature has on the overall model. To have a fully functional system, one has to be able to obtain the "proper" values for the α_j parameters. These weights are estimated during the training phase to maximize the likelihood of the training data (Berger et al. 1996). Given these weights, the model computes the probability distribution over labels for a particular example x as follows:

$$P(y \mid x) = \frac{1}{Z(x)} exp\left(\sum_{j=1}^{m} \alpha_j f_j(x,y)\right) \tag{9.1}$$

$$Z(x) = \sum_{y'} exp\left(\sum_{j=1}^{m} \alpha_j f_j(x,y')\right) \tag{9.2}$$

where $Z(x)$ is a normalization factor.

By using a training data of L examples, we are looking to find the parameter set that maximizes the log-likelihood of the data:

$$\log \prod_{i=1}^{L} p(y_i \mid x_i) = \sum_{i=1}^{L} \log p(y_i \mid x_i) \tag{9.3}$$

in other words, we are looking for the solution to the problem

$$\hat{p} = \arg\max_{\alpha} \sum_{i=1}^{L} \log p_\alpha (y_i \mid x_i) \tag{9.4}$$

[1] A binary feature f_j is merely an indicator function $f_j : \mathcal{X} \to \{0,1\}$.

Since the value in Eq. (9.3) is a convex function of the parameters $\{\alpha_j\}$, finding the exponential model that has the maximum data likelihood becomes a classical optimization problem, which has a unique solution. There have been several methods proposed to find such an optimum point, such as *generalized iterative scaling* (GIS) (Darroch and Ratcliff 1972), *improved iterative scaling* (IIS) (Berger et al. 1996), the limited-memory *Broyden-Fletcher-Goldfarb-Shanno* approximate gradient procedure (BFGS) (Liu and Nocedal 1989), and *sequential conditional generalized iterative scaling* (SCGIS) (Goodman 2002).[2]

While the MaxEnt method can nicely integrate multiple feature types seamlessly, in certain cases it can overestimate its confidence in especially low-frequency features. Let us clarify through an example: let's assume that we are interested in computing the probability of getting heads while tossing a (possibly unbiased) coin, and let's assume that we tossed it four times and got heads four times and no tails. Furthermore, let's assume that our model has two features: $f_1(x,y) = y$ is *heads* and $f_2(x,y) = y$ is *tails*. Then our constraints would be that $E_{\hat{p}}[f_1] = 1$ and $E_{\hat{p}}[f_2] = 0$, which in turn will enforce that our model will always predict that heads will show up with probability 1, which is, of course, premature with only four tosses. The problem here comes from our enforcing a hard constraint on a feature whose estimation is not reliable enough. There are several adjustments that can be made to the model to address this issue, such as regularization by adding *Gaussian priors* (Chen and Rosenfeld 2000) or *exponential priors* (Goodman 2004) to the model, using *fuzzy MaxEnt boundaries* (Khudanpur 1995), or using *MaxEnt with inequality constraints* (Kazama and Tsujii 2003).

Of the various methods mentioned above for estimating the "optimal" α_j values, one that has shown itself to be both fast and robust for mention detection (as well as for a wide range of NLP problems) is the *sequential conditional generalized iterative scaling* (SCGIS) technique (Goodman 2002). To overcome the problem of overestimating confidence in low-frequency features especially, we recommend starting with the regularization method based on adding *Gaussian priors* as described in Chen and Rosenfeld (2000).[3] Intuitively, this measure will model parameters as being close to 0 in value, unless the data suggests they are not. After computing the class probability distribution, the chosen criterion is the one with the most a posteriori probability. The decoding algorithm, described in section "Search for Mentions", performs sequence classification, through dynamic programming.

Now let $x_1^L = (x_1, x_2, \ldots x_L)$ be a sequence of contiguous tokens (i.e., a sentence or a document). The goal of a mention detection system is to find the most likely sequence of token labels $y_1^L = (y_1, y_2 \ldots y_L)$ that best matches the input x_1^L. We use the Viterbi dynamic-programing search algorithm (Viterbi 1967) to compute the

[2]Describing these methods is beyond the scope of this chapter. Please refer to the cited material for in-depth descriptions.

[3]Note that the resulting model cannot really be called a maximum entropy model, as it does not yield the model which has the maximum entropy (the second term in the product), but rather is a maximum a posteriori model.

probability of the output sequence $y_1 \ldots y_L$ given the input sequence $x_1 \ldots x_L$. In our case of mention detection, and similar to the approach described in Florian et al. (2004) and Zitouni et al. (2009), each token x_i in x_1^L is tagged with a label y_i as follows[4]:

- If the token is not part of any mention (is "outside" of any mention) it is tagged with ***O***
- If the token is part of an entity, it obtains a complex tag with a prefix and a suffix; the prefix indicates whether the token begins a new mention (*B*-) or is inside a mention (*I*-), and the suffix corresponds to a mention type. For example, the tag *B-person* is used for the first token of a person mention. In the ACE task, there are seven such main entity types: person, organization, location, facility, geopolotical entity (gpe), weapon, and vehicle.

Search for Mentions

Now that we have our model, we are interested in using it to find the mentions in a sentence. These mentions have strong interdependencies which cannot be properly modeled if the classification is performed independently for each token. We view this problem as *sequence classification*, in contrast to an *example-based classification* problem: given a sequence of tokens in a sentence $x_1 x_2 \ldots x_L$, our goal is to assign tags (labels) to each token, resulting in a sequence of tags $y_1 y_2 \ldots y_L$. One approach would be to take each example to be classified as the entire sequence of tokens, that is, the entire sentence. Such a space has a very high dimensionality, and we would run very soon into data sparseness problems. Instead, we will apply the Markov assumption, which states that the tag associated with the token i depends only on the tags associated with the tokens at positions $i-k+1 \ldots i-1$, where k has usually a value equal to 2 or 3. Given this assumption, and the notation $x_1^L = x_1 \ldots x_L$, the conditional probability of assigning the tag sequence y_1^L to the token sequence x_1^L becomes

$$p\left(y_1^L \mid x_1^L\right) = p\left(y_1 \mid x_1^L\right) p\left(y_2 \mid x_1^L, y_1\right) \ldots p\left(y_L \mid x_1^L, y_{L-k+1}^{L-1}\right) \tag{9.5}$$

and our goal is to find the sequence that maximizes this conditional probability

$$\hat{y}_1^L = \arg\max_{y_1^L} p\left(y_1^L \mid x_1^L\right) \tag{9.6}$$

[4]The mention encoding is the iob2 encoding presented in Tjong Kim Sang and Veenstra (1999) and introduced by Ramshaw and Marcus (1994) for base noun phrase chunking.

While we restricted the conditioning on the classification tag sequence to the previous k tags, we do not impose any restrictions on the conditioning on the tokens – the probability is computed using the entire token sequence x_1^L. In practical situations, though, features will only examine a limited context of the particular token of interest, but they are allowed to "look ahead", i.e. to examine features of the tokens succeeding the current token.

Under the constraint described in Eq. (9.5), the sequence in Eq. (9.6) can be efficiently identified. To obtain it, we create a *classification tag lattice* (also called *trellis*), as follows:

- Let x_1^L be the token input sequence and $S = \{s_1, s_2, \ldots, s_m\}$ be an enumeration of $\mathcal{Y}^k$ ($m = |\mathcal{Y}|^k$). We will call an element s_j a state. Every such state corresponds to the labeling of k successive tokens. We find it useful to think of an element s_i as a vector with k elements. We will use the notations $s_i[j]$ for j^{th} element of such a vector (the label associated with the token $x_{i-k+j+1}$) and $s_i[j_1 \cdots j_2]$ the sequence of elements between indices j_1 and j_2.
- We conceptually associate every character $x_i, i = 1, \ldots, L$ with a copy of S, $S^i = \{s_1^i, \ldots, s_m^i\}$; this set represents all the possible labeling of characters x_{i-k+1}^i at the stage where x_i is examined.
- We then create links from the set S^i to the S^{i+1}, for all $i = 1 \ldots L-1$, with the property that

$$w\left(s_{j_1}^i, s_{j_2}^{i+1}\right) = \begin{cases} p\left(s_{j_1}^{i+1}[k] \mid x_1^L, s_{j_2}^{i+1}[1..k-1]\right) \\ \quad if\ s_{j_1}^i[2..k] = s_{j_2}^{i+1}[1..k-1] \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

These weights correspond to probability of a transition from the state $s_{j_1}^i$ to the state $s_{j_2}^{i+1}$. If the states are not compatible (i.e. there is no possible tag sequence Y such that $Y[i-k+1, \ldots, i]$ is the sequence of labels associated with the tokens x_{i-k+1}^i and $Y[i-k+2, i+1]$ is the sequence of classification tags associated with the token sequence x_{i-k+2}^{i+1}), then the weight is 0. If the two states are compatible, the weight is proportional to predicting the tag $s_{j_2}^{i+1}[k]$ in the tag context $s_{j_2}^{i+1}[1 \ldots k-1]$ and observed token sequence x_1^L.

- For every token x_i, we compute recursively.[5]

$$\alpha_0\left(s_j\right) = 0, j = 1, \ldots, k$$
$$\alpha_i\left(s_j\right) = \max_{j_1 = 1, \ldots, M} \alpha_{i-1}\left(s_{j_1}\right) + \log w\left(s_{j_1}^{i-1}, s_j^i\right)$$
$$\gamma_i\left(s_j\right) = \arg\max_{j_1 = 1, \ldots, M} \alpha_{i-1}\left(s_{j_1}\right) + \log w\left(s_{j_1}^{i-1}, s_j^i\right)$$

[5]For convenience, the index i associated with state s_j^i is moved to α; the function $\alpha_i\left(s_j\right)$ is in fact $\alpha\left(s_j^i\right)$.

Intuitively, $\alpha_i(s_j)$ represents the log-probability of the most probable path through the lattice that ends in state s_j after i steps, and $\gamma_i(s_j)$ represents the state just before s_j on that particular path.[6]

- Having computed the $(\alpha_i)_i$ values, the algorithm for finding the best path, which corresponds to the solution of Eq. (9.6) is
  1. Identify $\hat{s}_L^L = \arg\max_{j=1\ldots L} \alpha_L(s_j)$
  2. For $i = L-1\ldots 1$, compute $\hat{s}_i^i = \gamma_{i+1}(\hat{s}_{i+1}^{i+1})$
  3. The solution for Eq. (9.6) is given by

$$\hat{y} = \left\{\hat{s}_1^1[k], \hat{s}_2^2[k], \ldots, \hat{s}_L^L[k]\right\}$$

The full algorithm is presented in Algorithm 1. The time complexity of the algorithm is $\Theta\left(|\mathcal{Y}|^k \cdot L\right)$, linear in the size of the sentence L but exponential in the size of the Markov dependency, k. To reduce the search space, we use *beam-search*.

Algorithm 1 Viterbi Search

Input: tokens w_1^L.
Output: the most probable sequence of tags (i.e., mentions) $\hat{y}_1^L = \arg\max_{y_1^L} P\left(y_1^L | x_1^L\right)$
Create $S = \{s_1, \ldots, s_M\}$, an enumeration of $\mathcal{Y}^k$
for $j = 1, M$ **do** $a_j \leftarrow 0$
for $i = 1-k, L+k$ **do**
  for $j = 1, M$ **do**
    $\gamma_{ij} = 1$, $b_j = -\infty$
    for $j' = 1, M$ **such that** $s_{j'}[2..k] = s_j[1..k-1]$ **do**
      $v \leftarrow a_{j'} - \log w\left(s_{j'}^{i-1}, s_j^i\right)$
      if $(v > b_j)$ **then**
        $b_j \leftarrow v$, $\gamma_{ij} \leftarrow j'$
  $a \leftarrow b$
$\hat{s}_{L+k} = \arg\max_{j=1\ldots m} a_j$
$j = \arg\max_j \gamma_{L+k,j}$
for $i = L+k-1\ldots 1$ **do** $\hat{s}_i \leftarrow s_j$, $j \leftarrow \gamma_{i+1,j}$
$\hat{y}_1^L \leftarrow (\hat{s}_1[1], \hat{s}_2[1], \ldots, \hat{s}_L[1])$

Beam Search

Anyone implementing Algorithm 1 faces a practical challenge: even for small values of k, the space $\mathcal{Y}^k$ can be quite large, especially if the classification space is large. This problem arises because the algorithm's search space size is proportional to $|\mathcal{Y}|^k$. This is the reason why in practice, for many natural language processing

[6] For numerical reasons, the values α_i are computed in log space, since computing them in normal space will result in underflow for even short sentences. Alternatively, one can compute a normalized version of the α_i coefficients, where they are normalized at each stage by the sum of all coefficients in the trellis column.

tasks, a *beam-search* algorithm is preferred instead. This algorithm is constructed around the idea that many of the nodes in the trellis have such small α-values that they will not be included in any "good" paths, and therefore can be skipped from computation without any loss in performance. To achieve this, the algorithm will keep only a few of the $M = |\mathcal{Y}|^k$ states alive at any trellis stage i. Then, after computing the expansion of those nodes for stage $i+1$, it eliminates some of the resulting states, based on their α_i values. One can use a variety of filtering techniques, among which the two most commonly used are:

- Using a fixed beam: keep only the n top-scoring candidates at each stage i for expansion
- Using a variable beam: keep only the candidates that are within a specified relative distance (in terms of α_i) from the top scoring candidate at stage i

Both options are good choices. Experience shows that one can use a beam of 5 and a relative beam of 30% to speed up the computation significantly (20–30 times) with almost no drop in performance. These parameter values should be optimized on a held-out development data set for each task, and may also vary depending on how one wants to trade off speed for accuracy.

Clean-Mention-Detection System: Standard Features

Our baseline mention detection system, denoted as the *Clean model*, uses featues that can be divided into the following categories:

1. **Lexical Features** The identity and the context of a current token x_i is clearly one of the most important features in predicting whether x_i is a mention or not (Florian et al. 2004). Lexical features are implemented as token n-grams spanning the current token, both preceding and following it. For a token x_i, token n-gram features will contain the previous $n-1$ tokens ($x_{i-n+1},\ldots x_{i-1}$) and the following $n-1$ tokens ($x_{i+1},\ldots x_{i+n-1}$). Setting n equal to 3 turned out to be a good choice.
2. **Gazetteer-based Features** The gazetteer-based features we use are computed on tokens. The gazetteers consist of several class of dictionaries: including person names, country names, company names, etc. Dictionaries contain single names such as `John` or `Boston`, and also phrases such as `Barack Obama`, `New York City`, or `The United States`. During both training and decoding, when we encounter in the text a token or a sequence of tokens that completely matches an entry in a dictionary, we fire its corresponding class.

The use of this framework to build MD systems for clean English text has given very competitive results at ACE evaluations (Florian et al. 2006). Trying other classifiers is always a good experiment, which we didn't pursue here for two reasons: first, the MaxEnt system used here is state-of-the-art, as proven in evaluations and competitions – while it is entirely possible that another system might get better results, we don't think the difference would be large. Second, we are interested in

ways of improving performance on noisy data, and we expect any system to observe similar degradation in performance when presented with unexpected input – showing results for multiple classifier types might very well dilute the message, so we stuck to one classifier type (Zitouni et al. 2009).

Enhancements for Robustness

As stated above, our goal is to avoid spans of characters which are not suitable for target-language MD, while minimizing impact on MD accuracy for target-language text. English is the initial target language for the initial experiments described here.

More specifically, our task is to process data automatically in any unpredetermined format from any source, during which we strive to avoid outputting spurious mentions on:

- Non-language material likely mistakenly submitted to the MD system, including mark-up tags and other data-set detritus, as well as non-text data such as code or binaries,
- Non-target-character-set material, here, non-Latin-alphabet material, such as Arabic and Chinese in their native character sets, and
- Target-character-set material not in the target language, here, Latin-alphabet languages other than English.

It is important to note that this is not merely a document-classification problem; this non-target data is often closely interspersed with valid input text. Mark-up is the obvious example of interspersing; however, other categories of non-target data can also interleave tightly with valid input. A few examples:

- English text is sometimes infixed right in a Chinese sentence, such as 其他BBC网站
- Some translation algorithms will leave unchanged an untranslatable word, or will transliterate it into the target language using a character convention which may not be a standard known to the MD engine, and
- Some target-alphabet-but-non-target-language material will be compatible with the target language, particularly people's names. An example with English as the target language is `Barack Obama` in the Spanish text ...`presidente de Estados Unidos, Barack Obama, dijo el da 24 que`....

Therefore, to minimize needless loss of processable material, a robustness algorithm ideally does a sliding analysis, in which, character-by-character or word-by-word, material may be deemed to be suitable to process. Furthermore, a variety of strategies will be needed to contend with the diverse nature of non-target material and the patterns in which it will appear among valid input.

To address these issues, the following is a summary of algorithmic enhancements to MD:

1. Detection of standard file formats, such as SGML, and associated detagging,
2. Segmentation of the file into target- vs. non-target-character-set passages, with the latter not to be processed further,
3. Tokenization to determine word and sentence units, and
4. MD, augmented as follows:

   - Sentence-level categorization of likelihood of "clean" English.
   - If clean English was detected, run the same baseline model as described in section "Mention-Detection Algorithm".
   - If the text is determined to be a "bad" fit to English, run an alternate maximum-entropy model that is heavily based on gazetteers, using only context-independent (e.g. primarily gazetteer-based) features, intended to catch isolated obvious English or English-compatible names embedded in text which otherwise is foreign.
   - If the text is determined to be in between "clean" and "bad", use a "mixed" maximum-entropy MD model whose training data and feature set are augmented to handle interleaving of English with mark-up and other languages.

These MD-algorithm enhancements will be described in the following subsections.

Detection and Detagging for Standard File Formats

Some types of mark-up are well-known standards, such as SGML (Warmer and van Egmond 1989). Clearly the optimal way of dealing with them is to apply detectors of these specific formats, and associated detaggers, as done previously (Yi et al. 2003). For this reason, standard mark-up is not a subject of the current study; rather, our concern is with mark-up peculiar to specific data sets, as described above, and so while this step is part of our overall strategy, it is not explored in the present experiments.

Character-Set Segmentation

Some entity mentions may be recognizable in a non-target language which shares the target-language's character set, for example, a person's name recognizable by English speakers in an otherwise-not-understandable Spanish sentence. However, non-target character sets, such as Arabic and Chinese when processing English, represent pure noise for an IE system. Therefore, deterministic character-set segmentation is applied, to mark non-target-character-set passages to be avoided by the remainder of the system, or, in a multilingual system, to be diverted to a subsystem suited to process that character set. Characters which can be ambiguous with regard to character set, such as some punctuation marks, are attached to target-character-set

passages when possible, but are not considered to break otherwise-contiguous non-target-character-set passages surrounding them on both sides.

Tokenization

Subsequent processing is based on determination of the language of target-alphabet text. The fundamental unit of such processing is target-alphabet word, necessitating tokenization at this point into word-level units. This step includes punctuation separation as well as the detection of sentence boundaries (Zimmerman et al. 2006).

Robust Mention Detection

After the above preprocessing steps, we detect mentions using a cascaded approach that combines several MD classifiers. Our goal is to select among maximum-entropy MD classifiers trained separately to represent different degrees of "noisiness" occurring in many genres of data, including output of speech recognition and/or machine translation, informal communications, mixed-language material, varied forms of non-standard database mark-up, etc. We somewhat-arbitrarily choose to employ three classifiers as described below. We select a classifier based on a sentence-level determination of how well the material fits the target language, based on an *n*-gram language model built from clean target-language training text. This language model is used during decoding to compute the perplexity (*PP*) of each sentence, as a measure of its deviation from the model. The *PP* indicates the quality of the text in the target-language (i.e. English) (Brown et al. 1992); the lower the *PP*, the cleaner the text. A sentence with a *PP* lower than a threshold θ_1 is deemed to be "clean" and hence the "clean" baseline MD model described in section "Mention-Detection Algorithm" is used to detect mentions of this sentence. The clean MD model has access to standard features described in section "Clean-Mention-Detection System: Standard Features". In the case where a sentence looks particularly badly matched to the target language, defined as $PP > \theta_2$, we use a "*gazetteer-based*" MD model based on a dictionary look-up to detect mentions, thus retreating to seeking known mentions in a context-independent manner reflecting that most of the context consists of out-of-vocabulary words. In the case of an in-between determination, that is, a sentence with $\theta_1 < PP < \theta_2$, we use a "*mixed*" MD model, based on augmenting the training data set and the feature set as described in the next section. The values of θ_1 and θ_2 are estimated empirically on a separate development data set that is also used to tune the Gaussian prior (Chen and Rosenfeld 2000). This set contains a mix of clean English and Latin-alphabet-but-non-English text that is not used for traning and evaluation.

The advantage of this combination strategy is that we do not need pre-defined knowledge of the text source in order to apply an appropriate model. The selection

of the appropriate model to use for decoding is done automatically based on *PP* value of the sentence. The cost of this approach clearly is that it requires the added overhead of employing three MD models and the language model. We will show in the experiments section how this combination strategy is effective not only in maintaining good performance on a clean English text but also in improving performance on non-English data when compared to other source-specific MD models.

Gazetteer-Based Mention-Detection Model

For content which is a very poor match to the target language, this model endeavors to find isolated phrases which are either in the target language, or are compatible with it, for example, the name of a person or place which is the same in the target language and the primary language of the content. To this end, here we employ a model which does not attempt to use context. Therefore, the gazetteer-based MD model simply employs a subset of the clean model's features, restricting to the features reflecting gazetteer information excluding lexical context, reflecting the likelihood that in this poorly-modeled material, words surrounding any recognizable mention are foreign and therefore unusable.

Mixed Mention-Detection Model

The mixed MD model is designed to process "sentences" mixing English with non-English, whether foreign-language or non-language material. Our approach is to augment model training compared to the clean baseline by adding non-English, mixed-language, and non-language material, and to augment the model's feature set with both language-identification features more localized than the sentence-level perplexity described above, and other features designed primarily to distinguish non-language material such as mark-up codes.

Language-Identification Features

We apply an *n*-gram-based language classifier (Prager 1999) to variable-length sliding windows as follows. For each word, we run 1- through 6-preceding-word windows through the classifier, and 1- through 6-word windows beginning with the word, for a total of 12 windows, yielding for each window a result like:

```
0.235 Swedish
0.148 English
0.134 French
...
```

For each of the 12 results, we extract three features: the identity of the top-scoring language, here, Swedish; the confidence score in the top-scoring language, here, 0.235; and the score difference between the target language (English for these experiments) and the top-scoring non-target language, here, $0.148 - 0.235 = -0.087$. Thus we have a 36-feature vector for each word. We bin these and use them as input to a maximum-entropy classifier (separate from the MD classifier) which outputs "English" or "Non-English", and a confidence score. These scores in turn are binned into six categories to serve as a "how-English-is-it" feature in the augmented (mixed) MD model. The language-identification classifier and the maximum-entropy "how-English" classifier are each trained on text data separate from each other and from the training and test sets for MD. Note that these two classifiers serve as two successive stages of determination of the quality of English text; the former at a broad level, one sentence at a time, to choose a classifier, and the latter at a narrower level of a word window, to serve as a feature for finer-grained analysis within sentences determined by the former to be mid-quality.

Additional Features

The following features are designed to capture evidence of whether a "word" is in fact linguistic material or not: number of alphabetic characters, number of characters, maximum consecutive repetitions of a character, numbers of non-alphabetic and non-alphanumeric characters, fraction of characters which are alphabetic, fraction alphanumeric, and number of vowels. These features are part of the augmentation of the mixed MD model relative to the clean MD model.

Data Sets

Four data sets are used for our initial experiments. One, "English", consists of 367 documents totaling 170,000 words, drawn from web news stories from various sources and detagged to be plain text. This set is divided into 340 documents as a training set and 27 for testing, annotated as described in more detail elsewhere (Han 2010). These data average approximately 21 annotated mentions per 100 words.

The second set, "Latin", consists of 23 detagged web news articles from 11 non-English Latin-alphabet languages totaling 31,000 words. Of these articles, 12 articles containing 19,000 words are used as a training set, with the remaining used for testing, and each set containing all 11 languages. They are annotated using the same annotation conventions as "English", and from the perspective of English; that is, only mentions which would be clear to an English speaker are labeled, such as `Barack Obama` in the Spanish example in section "Enhancements for Robustness". For this reason, these data average only approximately 5 mentions per 100 words.

The third, "Transactions", consists of approximately 60,000 words drawn from a text data set logging real financial transactions. Figure 9.1 shows example passages from this database, anonymized while preserving the character of the content.

This data set logs transactions by a staff of customer-service representatives. English is the primary language, but owing to international clientele, occasionally representatives communicate in other languages, such as the German here, or in English but mentioning institutions in other countries, here, a Czech bank. Interspersed among text are codes specific to this application which delineate and identify various information fields and punctuate long passages. The application also places constraints on legal characters, leading to the unusual representation of underline and the "at" sign as shown, making for an e-mail address which is human-readable but likely not obvious to a machine. Abbreviations represent terms particularly common in this application area, though they may not be obvious without adapting to the application; these include standards like `HUF`, a currency code which stands for Hungarian forint, and financial-transaction peculiarities like `BNF` for "beneficiary" as seen in Fig. 9.1. In short, good English is interspersed with non-language content, foreign-language text, and rough English like data-entry errors and haphazard abbreviations. These data average 4 mentions per 100 words.

Data sets with peculiarities analogous to those in this Transactions set are commonplace in a variety of settings. Training specific to data sets like this is often infeasible due to lack of labeled data, insufficient data for training, and the multitude of such data formats. For this reason, we do not train on Transactions, letting our testing on this data set serve as an example of testing on such data formats unseen.

Experiments

MD systems were trained to recognize the 116 entity-mention types shown in Table 9.1, annotated as described previously (Han 2010). The clean-data classifier was trained on the English training data using the feature set described in section "Clean-Mention-Detection System: Standard Features". The classifier for "mixed"-quality data and the "gazetteer" model were each trained on that set plus the "Latin" training set and the supplemental set. The fourth "supplemental" data set is added to training for some experiments, for purposes of robustness. This set consists of 42,000 words of varying degrees of imperfection as English: outputs of a system for machine-translation into English, Spanish text, and source and binary computer code. In addition, "mixed" training included the additional features described in section "Mixed Mention-Detection Model". The framework used to build the baseline MD system is similar to the one we used in the ACE evaluation.[7] This system has achieved competitive results with an *F*-measure of 82.7 when trained on the seven main types of ACE data with access to wordnet and part-of-speech-tag information as well as output of other MD and named-entity recognizers (Zitouni and Florian 2008).

[7]NIST's ACE evaluation plan: http://www.nist.gov/speech/tests/ace/index.htm.

Table 9.1 Entity-type categories used in these experiments. The eight in the right-most column are not further distinguished by mention type, while the remaining 36 are further classified as named, nominal or pronominal, for a total of $36 \times 3 + 8 = 116$ mention labels

Age	Event-custody	Facility	People	Date
Animal	Event-demonstration	Food	Percent	Duration
Award	Event-disaster	Geological-object	Person	E-mail-address
Cardinal	Event-legal	Geo-political	Product	Measure
Disease	Event-meeting	Law	Substance	Money
Event	Event-performance	Location	Title-of-a-work	Phone-number
Event-award	Event-personnel	Ordinal	Vehicle	Ticker-symbol
Event-communication	Event-sports	Organ	weapon	Time
Event-crime	Event-violence	Organization	Web-address	

Table 9.2 Performance of clean, mixed, and gazetteer-based mention detection systems as well as their combination. Performance is presented in terms of Precision (P), Recall (R), and *F*-measure (F)

	English			Latin			Transactions		
	P	R	F	P	R	F	P	R	F
Clean	78.7	73.6	76.1	16.0	40.0	22.9	19.5	32.2	24.3
Mixed	77.9	69.7	73.6	78.5	55.9	65.3	37.1	47.8	41.7
Gazetteer	76.9	66.2	71.1	77.8	55.5	64.8	36.5	47.5	41.3
Combination	78.1	73.2	**75.6**	80.4	56.0	**66.0**	38.5	49.1	**43.2**

It is instructive to evaluate on the individual component systems as well as the combination, despite the fact that the individual components are not well-suited to all the data sets, for example, the mixed and gazetteer systems being a poorer fit to the English task than the baseline, and vice versa for the non-target data sets. Precision/recall/*F*-measure results are shown in Table 9.2. Not surprisingly, the baseline system, intended for clean data, performs poorly on noisy data. The mixed and gazetteer systems, having a variety of noisy data in their training set, perform much better on the noisy conditions, particularly on Latin-alphabet-non-English data because that is one of the conditions included in its training, while Transactions remains a condition not covered in the training set and so shows less improvement. However, because the mixed classifier, and moreso the gazetteer classifier, are oriented to noisy data, on clean data they suffer in performance by 2.5 and 5 *F*-measure points, respectively. But system combination serves well, by recovering all but 0.5 *F*-measure point of this loss, while also actually performing better on the noisy data sets than the two classifiers specifically targeted toward them, as can be seen in Table 9.2. It is important to note that the combination model provides a major advantage: it frees MD from the need for pre-determination of the type of data source to select a model. With the combination model, we assume that the data source is unknown, which is our claim in this work, and we show that we obtain better performance than using source-specific MD models. This reflects the fact that a noisy data set will in fact have portions with varying degrees of "noise", so the combination outperforms any single model targeted to a single particular level of noise, enabling the system to contend with such variability without the need for pre-segregating sub-types of data for noise level. The obtained improvement from the system combination over all other models is statistically significant based on the stratified bootstrap re-sampling significance test (Noreen 1989). We consider results statistically significant when $p < 0.05$, which is the case in this paper. This approach was used in the named-entity-recognition shared task of CoNLL-2002.[8]

The great contribution to accuracy provided by the mixed model raises the question of the relative contribution of added data and added features to this model as compared to the clean model. To address this, we pursued a new experimental condition in which we attempt via further data supplementation to replace the gains

[8] http://www.cnts.ua.ac.be/conll2002/ner/.

Table 9.3 Performance of clean, mixed, and gazetteer-based mention detection systems as well as their combination. Performance is presented in terms of Precision (P), Recall (R), and *F*-measure (F)

	English			Latin			Transactions		
	P	R	F	P	R	F	P	R	F
Clean	78.7	73.6	76.1	16.0	40.0	22.9	19.5	32.2	24.3
Clean + MoreData	78.9	72.5	75.6	79.5	61.8	69.5	37.1	48.3	42.0
Clean + MoreData + EngVocab	78.8	72.2	75.4	81.2	62.1	70.3	38.2	49.8	43.3
Combination1	79.3	72.8	75.9	82.9	60.6	70.0	38.8	50.5	43.9

made by feature engineering. To this end, we add the "supplemental" set to the training data, but then train a model using the same feature set as the "clean" model. Table 9.3 shows that this condition, "Clean + MoreData" meets or exceeds the performance of "Mixed" on each category of data. "Clean + MoreData + EngVocab" is a related condition in which model training extracts its vocabulary only from the English training data, and "Combination1" is analogous to the original "Combination" model but with the Mixed-model replaced by the Mixed + MoreData model.

It should be noted that some completely-non-target types of data, such as non-target-character-set data, have been omitted from analysis here. Including them would make our system look comparatively stronger, as they would have only spurious mentions and so generate false alarms but no correct mentions in the baseline system, while our system deterministically removes them.

As mentioned above, we view MD robustness primarily as an effort to eliminate, relative to a baseline system, large volumes of spurious "mentions" detected in non-target input content, while minimizing disruption of detection in target input. A secondary goal is recall in the event of occasional valid mentions in such non-target material. Thus, as input material degrades, precision increases in importance relative to recall. As such, we view precision and recall asymmetrically on this task, and so rather than evaluating purely in terms of *F*-measure, we perform a detection-error-trade-off (DET) (Martin et al. 1997) analysis, in which we plot a curve of miss rate on valid mentions vs. false-alarm rate, with the curve traced by varying a confidence threshold across its range. We measure false-alarm and miss rates relative to the number of actual mentions annotated in the data set:

$$\text{FA rate} = \frac{\text{\# false alarms}}{\text{\# annotated mentions}} \tag{9.7}$$

$$\text{Miss rate} = \frac{\text{\# misses}}{\text{\# annotated mentions}} \tag{9.8}$$

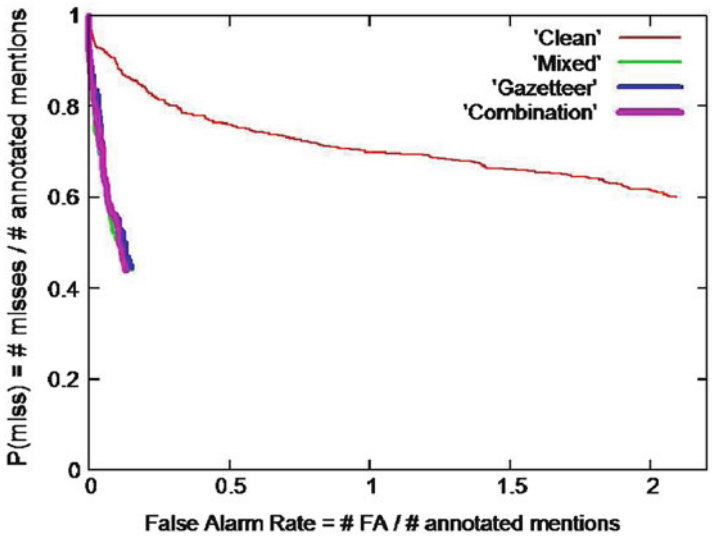

Fig. 9.2 Latin-alphabet-non-English data: DET plot for clean (*baseline*), mixed, gazetteer, and combination MD systems on the Latin-alphabet-non-English text. The clean system (*upper curve*) performs far worse than the other three systems designed to provide robustness; these systems in turn perform nearly indistinguishably

where false alarms are "mentions" output by the system but not appearing in annotation, while misses are mentions which are annotated but do not appear in the system output. Each mention is treated equally in this analysis, so frequently-recurring entity/mention types weigh on the results accordingly.

Figure 9.2 shows a DET plot for the clean, mixed, gazetteer, and combination systems on the "Latin" data set, while Fig. 9.3 shows the analogous plot for the "Transactions" data set. The drastic gains made over the baseline system by the three experimental systems are evident in the plots. For example, on Latin, choosing an operating point of a miss rate of 0.6 (nearly the best achievable by the clean system), we find that the robustness-oriented systems eliminate 97% of the false alarms of the clean baseline system, as the plot shows false-alarm rates near 0.07 compared to the baseline's of 2.08. Gains on Transaction data are more modest, owing to this case representing a data genre not included in training. It should be noted that the jaggedness of the Transaction curves traces to repetition of some of the terms in this data set.

In making a system more oriented toward robustness in the face of non-target inputs, it is important to quantify the effect of these systems being less-oriented toward clean, target-language text. Figure 9.4 shows the analogous DET plot for the English test set, showing that achieving robustness through the combination system comes at a small cost to accuracy on the text the original system is trained to process.

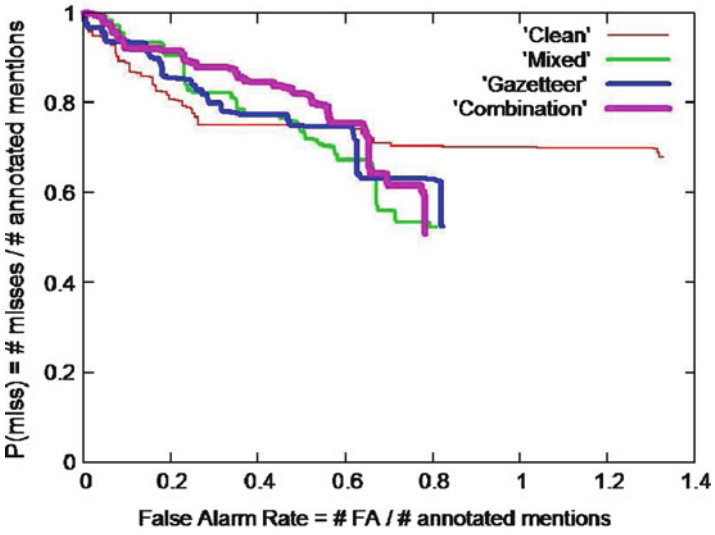

Fig. 9.3 Transactions data: DET plot for clean (*baseline*), mixed, gazetteer, and combination MD systems on the Transactions data set. The clean system (*upper/longer curve*) reaches far higher false-alarm rates, while never approaching the lower miss rates achievable by any of the other three systems, which in turn perform comparably to each other

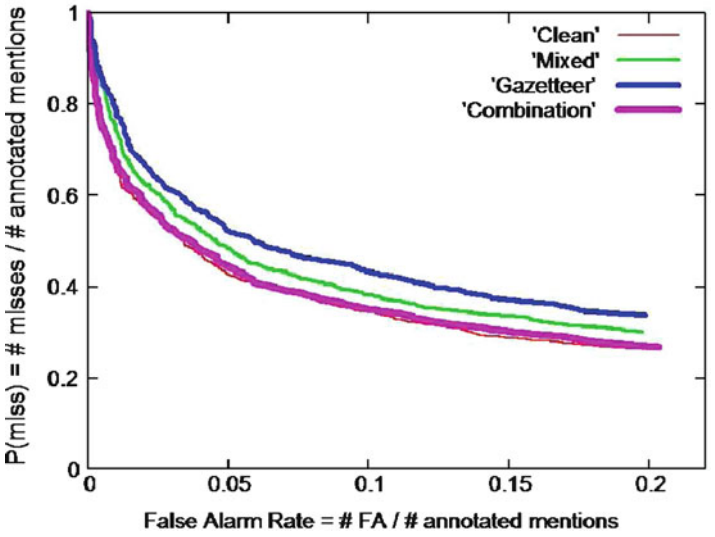

Fig. 9.4 DET plot for clean (*baseline*), mixed, gazetteer, and combination MD systems on clean English text, verifying that performance by the clean system (*lowest curve*) is very closely approximated by the combination system (*second-lowest curve*), while the mixed system performs somewhat worse and the gazetteer system (*top curve*), worse still, reflecting that these systems are increasingly oriented toward noisy inputs

Summary

For information-extraction systems to be useful, their performance must degrade gracefully when confronted with inputs which deviate from ideal and/or derive from unknown sources in unknown formats. Imperfectly-recognized speech, and imperfectly-translated, mixed-language, marked-up text and non-language material must not be processed in a garbage-in-garbage-out fashion merely because the system was designed only to handle clean material in one language. Thus we have embarked on information-extraction-robustness work, to improve performance on imperfect inputs while minimizing disruption of processing of clean natural-language input. We have demonstrated that for one IE task, mention detection, a multi-faceted approach, motivated by the diversity of input data imperfections, can eliminate a large proportion of the spurious outputs compared to a system trained on the target input. Such robustness comes at a relatively small cost of accuracy on clean input. This outcome is achieved by a system-combination approach in which a perplexity-based measure of how well the input matches the target language is used to select among models designed to deal with such varying levels of noise. Rather than relying on explicit recognition of genre of source data, the experimental system merely does its own assessment of how much each sentence-sized chunk matches the target language, an important feature in the case of unknown sources.

Chief among directions for further work are to continue to improve performance on noisy data, and to strengthen our findings via larger data sets. We should explore using the PP and the "how-English" criteria to filter the data to be used to train the "clean" model. Additionally, we look forward to expanding analysis to different types of imperfect input, such as machine-translation output, different types of mark-up, and different genres of content. Further work should also explore the degree to which the approach to achieving robustness must vary according to the target language. Finally, robustness work should be expanded to other information-extraction in addition to mention detection.

References

Benajiba Y, Rosso P (2008) Arabic named entity recognition using conditional random fields. In: Workshop on HLT and NLP within the Arabic world. Arabic language and local languages processing: status updates and prospects, 6th international conference on language resources and evaluation – LREC-2008, Marrakech

Benajiba Y, Diab M, Rosso P (2009) Arabic named entity recognition: a feature-driven study. IEEE Trans Audio Speech and Lang process 17:926. In the special issue on processing morphologically rich languages

Berger A, Della Pietra S, Della Pietra V (1996) A maximum entropy approach to natural language processing. Comput Linguist 22(1):39–71

Bikel DM, Miller S, Schwartz R, Weischedel R (1997) Nymble: a high-performance learning name-finder. In: Proceedings of ANLP-97, Washington, DC, pp 194–201

Borthwick A, Sterling J, Agichtein E, Grishman R (1998) Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In: Proceedings of the sixth workshop on very large corpora, pp 152–160

Brown PF, Della Pietra SA, Della Pietra VJ, Lai JC, Mercer RL (1992) An estimate of an upper bound for the entropy of english. Comput Linguist 18(1):31–40

Chen S, Rosenfeld R (2000) A survey of smoothing techniques for ME models. IEEE Trans Speech Audio Process 18:37–50

Chieu H-L, Tou Ng H (2003) Named entity recognition with a maximum entropy approach. In: Conference on computational natural language learning: CoNLL-2003, Edmonton

Dan Klein HN, Smarr J, Manning C (2003) Named entity recognition with character-level models. In: Conference on computational natural language learning: CoNLL-2003, Edmonton

Darroch JN, Ratcliff D (1972) Generalized iterative scaling for log-linear models. Ann Math Stat 43(5):1470–1480

Florian R, Hassan H, Ittycheriah A, Jing H, Kambhatla N, Luo X, Nicolov N, Roukos S (2004) A statistical model for multilingual entity detection and tracking. In: Proceedings of HLT-NAACL 2004, Los Angeles, pp 1–8

Florian R, Ittycheriah A, Jing H, Zhang T (2003) Named entity recognition through classifier combination. In: Conference on computational natural language learning: CoNLL-2003, Edmonton

Florian R, Jing H, Kambhatla N, Zitouni I (2006) Factorizing complex models: a case study in mention detection. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Morristown, Sydney, Australia, pp 473–480

Goodman J (2002) Sequential conditional generalized iterative scaling. In: Proceedings of ACL'02, Philadelphia

Goodman J (2004) Exponential priors for maximum entropy models. In: Marcu D, Dumais S, Roukos S (eds) HLT-NAACL 2004: main proceedings. Association for Computational Linguistics, East Stroudsburg, Boston, MA, pp 305–312

Han DB (2010) Klue annotation guidelines – version 2.0. Technical Report RC25042, IBM Research

Kazama J, Tsujii J (2003) Evaluation and extension of maximum entropy models with inequality constraints. In: Collins M, Steedman M (eds) Proceedings of the 2003 conference on empirical methods in natural language processing, Sapporo, pp 137–144

Khudanpur S (1995) A method of maximum entropy estimation with relaxed constraints. In: 1995 Johns Hopkins University language modeling workshop, Johns Hopkins University, Baltimore, Maryland

Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. Math Program 45(3(Ser. B)), 503–528

Martin A, Doddington G, Kamm T, Ordowski M, Przybocki M (1997) The det curve in assessment of detection task performance. In: Proceedings of the European conference on speech communication and technology (Eurospeech), Rhodes, pp 1895–1898

Miller D, Boisen S, Schwartz R, Stone R, and Weischedel R (2000) Named entity extraction from noisy input: speech and ocr. In: Proceedings of the sixth conference on applied natural language processing. Association for computational linguistics, Morristown, pp 316–324

Minkov E, Wang RC, Cohen WW (2005) Extracting personal names from email: applying named entity recognition to informal text. In: Proceedings of human language technology conference and conference on empirical methods in natural language processing, pp 443–450, Vancouver, British Columbia. Association for Computational Linguistics

Noreen EW (1989) Computer-intensive methods for testing hypotheses. Wiley, New York

Pitrelli JF, Lewis BL, Epstein EA, Franz M, Kiecza D, Quinn JL, Ramaswamy G, Srivastava A, Virga P (2008) Aggregating distributed STT, MT, and information extraction engines: the GALE interoperability-demo system. In: Interspeech, Brisbane

Prager JM (1999) Linguini: language identification for multilingual documents. J Manage Inf Syst 16(3):71–101

Ramshaw L, Marcus M (1994) Exploring the statistical derivation of transformational rule sequences for part-of-speech tagging. In: Proceedings of the ACL workshop on combining symbolic and statistical approaches to language, Las Cruces, Las Crises, New Mexico, pp 128–135

Ramshaw L, Marcus M (1999) Text chunking using transformation-based learning. In: Armstrong S, Church KW, Isabelle P, Manzi S, Tzoukermann E, Yarowsky D (eds) Natural language processing using very large Corpora. Kluwer, Dordrecht, pp 157–176

Tjong Kim Sang EF (2002) Introduction to the conll-2002 shared task: language-independent named entity recognition. In: Proceedings of CoNLL-2002, Taipei, pp 155–158

Tjong Kim Sang EF, Veenstra J (1999) Representing text chunks. In: Proceedings of EACL'99, Bergen

Tri Tran Q, Thao Pham TX, Hung-Ngo Q, Dinh D, Collier N (2007) Named entity recognition in vietnamese documents. Journal Progress in Informatics 4:3–13

Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Trans Inf Process 13:260–269

Warmer J, van Egmond S (1989) The implementation of the amsterdam sgml parser. Electron Publ Origin Dissem Des 2(2):65–90

Yi L, Liu B, Li X (2003) Eliminating noisy information in web pages for data mining. In: KDD '03: proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, pp 296–305

Zimmerman M, Hakkani-Tur D, Fung J, Mirghafori N, Gottlieb L, Shriberg E, Liu Y (2006) The icsi+multilingual sentence segmentation system. In: Interspeech, Pittsburgh, pp 117–120

Zitouni I, Florian R (2008) Mention detection crossing the language barrier. In: Proceedings of EMNLP'08, Honolulu

Zitouni I, Florian R (2009) Cross-language information propagation for arabic mention detection. ACM Trans Asian Lang Inf Process (TALIP) 8(4):1–21

Zitouni I, Luo X, Florian R (2009) A cascaded approach to mention detection and chaining in arabic. IEEE Trans Audio Speech Lang Process 17:935–944

Chapter 10
A Name Is Worth a Thousand Pictures: Referential Practice in Human Interactions with Internet Search Engines

Robert J. Moore

Abstract Today's Internet search engines are highly effective in returning relevant web pages to users in fractions of seconds. Yet interactions with search engines are far from trouble free. When interacting with search engines, users experience a variety of troubles, which are still poorly understood. One particular kind of trouble stems from users' prior knowledge about entities of interest, particularly regarding their names. This study examines how referential practice is organized in the context of search-engine interactions. It finds that, as in conversation, users employ *naming* in their queries to refer to entities if they can. However, when they do not know the name, or a name fails, they attempt a *two-stage search:* first they search for the entity name, using *generic descriptions* combined with *image search*, and second, if the name is found, they formulate subsequent queries using that name. Computer interaction analysis is used to reveal formal features of users' referential practices from recordings of screen video with eye tracking and design recommendations for search engines are offered.

Introduction

Search engines are among the top-ten most-visited websites on the Internet (Hitwise 2010). On a typical day, more than half of all Internet users accesses a search engine, and this number has been steadily increasing over the past decade (Fallows 2008). Current search engines achieve astounding performance in enabling users to find highly relevant pages in fractions of seconds. However users' search behaviors and typical rates of success are still poorly understood (Aula et al. 2008; Hassan et al. 2010).

R.J. Moore, Ph.D. (✉)
Yahoo! Research, 4401 Great America Parkway, Santa Clara, CA 95054, USA
e-mail: bobmoore@yahoo-inc.com; eroombob@gmail.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_10,
© Springer Science+Business Media New York 2013

Users are generally on their own when deciding on the most effective strategies for searching the Internet, and they must learn through trial and error.

One of the most critical parts of a web searcher's task is the *formulation* of queries. The particular choice of words and symbols is highly consequential for the particular results that will be returned and for the possibility of success. The work of formulating queries includes decisions about the level of detail to include in the query, the number of words, the inclusion of definite articles, conjunctions and prepositions and more. Perhaps the most critical part of this formulation work is deciding how to *refer* to entities of interest. As in human conversation there are always multiple ways to refer to people, places and things. For example, "that sword," "a scimitar," "a curved sword" or "a sword like the guy in Indiana Jones" can all refer to the same thing. It is the task of the speaker or web searcher to decide *which* form of reference to use on any particular occasion given the particular recipient (Sacks and Schegloff 1979).

Choosing a form of reference in interaction is a function of both the speaker's knowledge and the presumed knowledge of the recipient. Speakers are constrained both by the names and descriptors they know and by the names and descriptors they suppose their recipients know. Similarly web searchers are constrained by their own knowledge of an entity and must make assumptions about the kinds of references a search engine can recognize. Like a human, a search engine is likely to recognize a proper name for an entity, but unlike a human it will not recognize more indexical references such as "the one I looked for last time" or "that one there."

While past work has examined the role of names in referential practice in conversation (Sacks and Schegloff 1979; Moore 2008), this study examines the role of names in the context of query formulation in Internet search. This study identifies a user practice not previously described in the search literature. The practice, *two-stage searches*, is employed on occasions in which users lack the name of the entity for which they are searching. Generic descriptors combined with image search can be used to compensate for the lack of a name, although with less precision and efficiency. In specifying the local organization of two-stage searches, this study uses computer interaction analysis (Moore et al. 2011; Moore and Churchill 2011), a method developed by the author and his colleagues, which adapts elements of ethnomethodology and conversation analysis to the study of human-computer interaction using eye-tracking video. Finally design implications for how search engines might better support the practice of "two-stage searches" are explored.

Background

Past research has tackled various aspects of the phenomenon of query formulation. For example past studies (Balasubramanian et al. 2010; Brendsky and Croft 2008) have examined the *length* of search queries and found that longer queries perform poorly compared to shorter, "keyword" queries. These studies then attempt to improve query performance through "query reduction," that is, automatically identifying and

removing or down-weighting "extraneous terms" from search queries. While such studies provide useful techniques for mechanically optimizing query performance, they provide little insight into *why* long queries perform poorly or into what makes a "keyword" query effective.

Other research on query formulation examines strategies users employ in *re*formulating queries. Rieh and Xie (2006) qualitatively examine sequences of queries and identify different patterns. For example, two such strategies include making a series of queries progressively *more general* or making them progressively *longer* while maintaining a *kernel* of keywords. Liu and Belkin (2008) take this approach further by identifying four reformulation patterns – simple term, specification, systematic expansion and limited expansion - and testing their relative performance. To their (Liu and Belkin 2008) surprise no significant differences in performance among these four reformulation strategies were found, and they conclude that the relationship between query reformulation and search outcome is "not as straightforward as expected." Liu and Belkin (2008) admit that one limitation of their approach is that they cannot infer users' intent.

Chai et al. (2007) examine a different kind of problem with query formulation: difficulty in formulating effective queries when searching for *images*. On the one hand, "content-based" approaches are difficult for users because users do not tend to formulate their queries in terms of low-level visual descriptors. On the other hand, "text-based" approaches, although more natural to users, suffer from the fact that the descriptions that accompany images are highly subjective and often absent. Like other verbose queries, descriptions of images tend to perform poorly as queries (Chai et al. 2007).

One limitation of all of the above studies of query formulation is that they tend to focus only on series of isolated queries across search sessions. As a result much of the *interactional context* – what happens in between submitted queries – gets stripped out of the data and lost to analysis.

A somewhat different approach has been conducted at Internet search companies themselves. In recent studies (Aula et al. 2010; Feild et al. 2010) industry researchers and academics examine the topic of "searcher frustration." Aula et al. (2010) bring users into the lab to observe directly embodied indicators of frustration including: "frowning, biting nails, leaning closer, and sighing." They then attempt to observe which behaviors correlate with these embodied signals and generate hypotheses about frustration for a large-scale log analysis. Analyzing series of submitted queries, like past studies, they (Aula et al. 2010) find that in unsuccessful sessions, users formulate more queries as questions, use more advanced operators, spend more time on search results pages and formulate their longest queries in the middle of the session. Similarly in the lab Feild et al. (2010) give users information-seeking tasks, ask them questions about their levels of frustration and record their embodied expressions of frustration with physical sensors. These sensors include a "mental state camera, pressure sensitive mouse and pressure sensitive chair" (Feild et al. 2010). They then use these data to model searcher frustration statistically and predict user reports of frustration.

Despite these sophisticated efforts, neither study (Aula et al. 2010; Feild et al. 2010) reveals a straightforward relationship between *physical* expressions of frustration and

observable searcher behaviors. In their results, Aula et al. (2010) make no mention of embodied indicators of frustration. They report no findings of the kind: queries formulated as questions are preceded by sighing and nail biting. Similarly, Feild et al. (2010) report a "surprise" that the physical sensors failed to "strongly correlate with user-reported frustration." The inability of these authors to discover a straightforward statistical relationship between "frustration" and particular activities or situations could be interpreted from an ethnomethodological perspective as evidence that *frustration* itself is not a singular cognitive state but rather a gloss that users (and researchers) use to refer to different kinds of interactional phenomena and especially troubles. This alternative approach seeks to discover particular user practices and troubles and specify their formal, sequential features. Taking such an approach, the present study focuses on a particular kind of interactional trouble that users sometimes face when using search engines, which they might characterize as "frustrating." It involves how users *refer* to the entities for which they search.

Referential practice involves the generic problem of how to choose a particular reference form given that there are always multiple ways to refer to the same entity. In the conversation analysis literature and mentioned above, Sacks and Schegloff (1979) examine this question with respect to references to persons in telephone calls, and they find two structural preferences: minimization and recipient design. That is, if possible, a person reference should use a single term, *and* it should be recognizable by this particular recipient. Sacks and Schegloff (1979) observe that *first names* satisfy both of these preferences, and they are used massively. However, Sacks and Schegloff (1979, p. 19) also observe what happens when both preferences cannot be met. Sometimes the recipient cannot be expected to recognize the name of the third person (e.g., Daniel). In such cases, the preference for minimization is relaxed just enough to enable the recipient to achieve recognition through combined references forms or descriptions (e.g., Daniel, the guy who cuts my grass). In short, speakers seek mutual recognition with the fewest words or least amount of interactional work possible. Furthermore, Sacks and Schegloff (1979, p. 17) observe that "…names are not only heavily used when known: they may be introduced for subsequent use when not already known to recipient, thereby arming him with the resources he may thereafter be supposed to have." In other words, speakers may *teach* their recipients the referent of a name so that they can use the minimal name throughout the remainder of the interaction.

Moore (2008) examines referential practice in the context of face-to-face service encounters, or rather, how customers at a copy shop refer to the document services they want when placing orders at a walk-up counter. Moore (2008) finds that as with references to absent third parties, references to (absent) document services tend to take the form of a *name*. In cases where there is no trouble, customers ask for things like "lamination," "scoring," "comb binding" and the like. But when customers encounter trouble with the use of a name or have no idea what to call the service, they tend to switch to visual *depictions* consisting of hand gestures and verbal descriptions. Such depictions require significantly more interactional work to achieve than the use of a name, but they are extremely flexible and can be effective as a last resort.

Methods and Data

This study employs *computer interaction analysis* (Moore and Churchill 2011; Moore et al. 2011) as its primary method. Computer interactions analysis enables detailed examination of the local, sequential organization of human interaction with a GUI-based computer through the use of eye-tracking, screen video and manual transcription. In contrast to past laboratory studies (Aula et al. 2010; Feild et al. 2010) and both qualitative and quantitative studies of search logs, computer interaction analysis enables the identification of users' practices in their rich contexts of interaction with a system. Through the recording of screen video and eye tracking, this approach offers a rich context for understanding users' intent as it is displayed in and through their concrete actions.

Inspired by Suchman (1987), computer interaction analysis adapts the basic approaches of ethnomethodology and conversation analysis to users' interactions with personal computers. While ethnomethodology (Garfinkel 1967) seeks to discover the procedures whereby human actors locally achieve order in concrete social settings, conversation analysis (Sacks et al. 1974) focuses more narrowly on how speakers locally organize *talk-in-interaction* through generic, but situated sequential practices. Both fields are concerned with discovering order in the concreteness of human action through direct observation.

The aim of computer interaction analysis (Moore and Churchill 2011) is to discover and specify the local, sequential organization of users' interaction with systems driven by a *graphical user interface* (GUI). Examination of rich records of user actions and machine behaviors, with the aid of a specialized transcription notation scheme, enables the qualitative analysis of the lived, situated work of human interaction with a computer. By empirically examining how users and computers organize their interactions, we can better understand user practices, identify misalignments between user and system, discover emergent user workarounds to interactional troubles and more. Moore et al. (2011) use computer interaction analysis to compare the organization of query repair in web search to repair in conversation. The current study design was motivated by this prior study in which self-generated search tasks were used. In that study users were instructed to generate a list of topics about which they were interested and then to define bounded search tasks from that list. While the technique generates "naturalistic" search behavior, in that the users search for information of genuine interest to them, it fails to generate many cases of *difficult searches*, especially those involving entities of which the users has no prior knowledge.

Therefore in the current study, special tasks were designed to elicit search troubles, somewhat like Aula et al. (2010) who design "difficult" search tasks. Seven participants, ranging in age from 23 to 45, were recorded performing 15 different tasks, which generated 101 analyzable video clips (4 were lost due to system crashes). Five users were male, two were female, and either held doctorate degrees or were enrolled in graduate school. The sample is thus highly educated and not necessarily representative of the general population in that respect.

The search tasks all involved people, places and things but did not provide their *names*. Entities were represented instead through photographs only (for example see Fig. 10.1).

Fig. 10.1 Devil's Tower. Devils Tower, Wyoming, USA (Picture taken by Colin Faulkingham in July 2005 and placed in the public domain. Accessed in Sept 2012 from http://en.wikipedia.org/wiki/File:Devils_Tower_CROP.jpg)

- *"Find a motel near this place"* + [photo of Devil's Tower, Wyoming]
- *"Find the train station nearest to this bridge"* + [photo of Jingu Bridge, Tokyo]
- *"Find a web page from which you can buy this product"* + [photo of a Dr. Martens Iconic Mary Jane shoe]

Each entity was relatively obscure but easy to find on the Internet with its name and was also distinctive enough that a purely descriptive query could find it with a little perseverance. Users were instructed to use *any* website they wanted and to stop the task when they believed they completed it or when they gave up. This procedure elicited multiple occasions on which users struggled with search tasks due to their lack of knowledge regarding its name. Therefore, the level of difficulty of the tasks was dependent on each participant's prior knowledge about the entities in question, *not* on the availability of the requested information on the Internet, as is the case with Aula et al.'s (2010) difficult search tasks. Yet an advantage of both kinds of tasks is that they enable the researcher to know the goals of the participants' search activities, which aids analysis.

The participants' behavior was captured through screen video, along with eye-gaze behavior, which was recorded using Tobii 1750 eye trackers (50 Hz sampling frequency). Detailed transcripts of interaction and gaze events were then constructed manually for sequences of interest, advancing video clips a frame at a time. Screen videos were recorded at 10 fps making each frame represent one-tenth of a second. A specialized notation scheme (Moore and Churchill 2011), inspired by that of Jefferson (2004), was used for transcribing users' input actions and eye-gaze behaviors, as well UI events on the system's display (see Transcript Notation at end of paper). Transcription enables fine-grained representation, especially of the *temporal* relationship among input, UI and eye-gaze events.

Fig. 10.2 Task success by form of reference with average number of queries per task

Form (avg. # queries)	Success	Failure
Name (1.4)	34	2
Related Name (1.8)	16	0
Description (6.7, 5.2/9.5)	31	18
TOTAL	81	20

Results

In order to complete the tasks in the study, participants had to refer to the entities depicted in the task images in some way. As expected, they employed names and descriptions in formulating their initial search queries (see Fig. 10.2). In one third of all cases, they used the specific name of the entity, whereas in nearly half, they used descriptions derived purely from the photographs and terms in the instructions. In other words, if users had prior knowledge of the name, they used it in the query, but if not, they relied on generic descriptions instead. With the *names* of the entities, participants completed the tasks quickly in 1.4 queries on average (one outlier thrown out). On the other hand, when the participants used only a generic description as their initial query, their search sessions expanded to 6.7 queries on average. If these latter sessions are divided into those that succeeded in completing the task and those that failed, we see that participants took 5.2 descriptive queries to complete the tasks and 9.5 before they gave up.

One surprise in the results was that in 16% of cases, participants employed a combination of names and descriptors. They produced a name, or a description that contained a name, that was *not* the name of the entity in the task image but rather was related to it in some way. Use of such related names were nearly as good as knowing the name of the entity: users completed the tasks in 1.8 queries. In the remainder of the paper, transcripts for search tasks involving each type of search strategy are examined in detail.

Names

In this study when the participants knew the name of the entity for which they were searching they used that name in their initial query. They did this in one-third of all cases. In about one-third of these cases involving the entities' names, participants

formulated their initial queries using the entity name plus additional task-relevant terms. For example (entity names italicized):

- *Flatiron building* cross streets
- Hotel *devil's tower*
- *Mini cooper* year first made
- Buy *doc martens mary jane*

In the remaining two-thirds of these name queries, participants' initial queries consisted of only the entity's name and were entered either in a search engine or a task-relevant site, such as IMDB.com (see below).

Excerpt 1 offers an example of a query containing the entity name plus task-relevant terms. In this case, the participant is trying to "Find a motel near this place" and is shown a photo of Devil's Tower National Monument in Wyoming (Fig. 10.1).

```
(1) [13-Place2]
001 <TYPES "hotel devil's tower", 2.0>
002 <ENTER> ("Connecting…" in page tab)
003 (0.3) {{looks down to where results will appear}}
004 //hotel devil's tower - Yahoo! Search Results//
005 (0.2) {{looks down to where results will appear}}
006 (results appear)
007 (0.4) {{Res#1 "Devils Tower Hotels"}}
008 (0.4) {{SponRes#2 "Devils Tower Hotels"}}
009 <SCROLL DOWN>
010 (0.3) {{Res#1 "Devils Tower Hotels"}}
011 <~MOUSE, NW>
012 (0.5) {{skims Res#1 "Devils Tower Hotels"}}
013 (0.1) {{SubRes#1 "Hotel Special Deals in Devils T"}}
014 (hovers over Res#1 "Devils Tower Hotels" title)
015 (0.6) {{reads SubRes#1 "Hotel Special Deals in Devils T"}}
016 (0.1) {{looks at Res#1 "Devils Tower Hotels"}}
017 (moves pointer off of Res#1 "Devils Tower Hotels" title)
018 (0.2) {{Res#1 "Devils Tower Hotels"}}
019 <.MOUSE, center of Res#1 snippet>
020 (0.3) {{Res#1 "Devils Tower Hotels"}}
021 <SCROLL DOWN>
022 (1.5) {{Res#2 "Devils Tower Hotels : Compare"}}
023 <~MOUSE, S>
024 (0.6) {{Res#2 "Devils Tower Hotels : Compare"}}
025 (hovers over Res#2 "Devils Tower Hotels : Compare Hotel")
026 (0.3) {{Res#2 "Devils Tower Hotels : Compare"}}
027 (0.5) {{Res#3 "Devils Tower Lodge, Devils Tower"}}
028 (moves pointer off of Res#2)
029 (0.7) {{Res#3 "Devils Tower Lodge, Devils Tower"}}
030 <.MOUSE, Res#3 "Devils Tower Lodge, Devils Tower"}}
031 <LEFT CLICK> (hand pointer turns black)
```

Fig. 10.3 Completing a task easily with the entity name (excerpt 1, line 31)

The participant's initial query, typed into the search box in the toolbar, consists of the name of the entity, "devil's tower," plus a task-relevant word, "hotel" (line 1). This returns a batch of results all of which contain the words "Devil's Tower" and references to accommodations (line 6, Fig. 10.3). The user briefly scans the first result, "Devils Tower Hotels" (line 07), followed by a brief scan of the second sponsored result (line 08). He then scrolls down (line 9), exposing the fourth result at the bottom of the screen (Fig. 10.3). The user then returns his gaze to the first result (line 10), moves his mouse toward it (line 11) and skims it for 0.5 s (line 12). Before the mouse pointer reaches the first result (line 14), the user's gaze continues down to read its indented, subordinate result, "Hotel Special Deals in Devils Tower – Find Cheap Hotel deals," (line 15). The user then looks back up at the first result (line 16) and scans it again (lines 18 and 20), while parking the mouse pointer over it but not on its title (lines 17 and 19). After inspecting the first result, the user scrolls down again (line 21) and inspects the second search result for 1.5 s (line 22). He moves the mouse pointer over the second result (lines 23 and 25) as he continues to scan it (lines 24 and 26). But again he does not select it. He continues on to inspect the third result, "Devils Tower Lodge, Devils Tower, WY - Yahoo! Travel," for 1.2 s (line 27 and 29) and hovers the pointer over that (line 30). This time he clicks the title link (line 31, Fig. 10.3) and completes the task. We see then an ideal case of a user finding a web page very quickly using an entity name. The user inspects the first three search results and finds an appropriate page within 12 s. This is an example of search engines at their best.

In the remaining two-thirds of cases involving name queries, initial queries consist of the entity's name and are submitted to either a search engine or task-relevant website:

- New york city *flatiron building* [Yahoo!]
- *Cooper mini* [Yahoo!]
- *Devil's tower* [wikipedia]
- *Cate blanchett* [IMDB]

In the following excerpt (2), the participant is trying to find the year the Mini Cooper automobile was first produced. Mini Cooper is of course referred to entirely with a photo of one. This participant exhibits recognition of the car by simply typing "cooper mini" in the search engine box in the toolbar (line 1).

```
(2) [14-Product2]
001 <TYPES "cooper mini", 2.5> {{eyes follow cursor}}
002 (0.4) {{"cooper mini"}}
003 <ENTER> ("Connecting…" appears in page tab)
004 (0.2) {{"cooper mini"}}
005 //cooper mini - Yahoo! Search Results//
006 (0.6) {{space above where results will appear}}
007 (results appear)
008 (0.4) {{glances down past sponsored results}}
009 <~MOUSE, S>
010 (0.3) {{Res#1 "MINI.com"}}
011 (0.3)
012 <.MOUSE, bottom center of Sponsored Results box>
013 <SCROLL DOWN 0.3>
014 <~MOUSE, W>
015 (0.7) {{Res#3 "Mini -Wikipedia, the free encyclop"}}
016 <.MOUSE, Res#3 "Mini -Wikipedia, the free encyclopedia">
017 <LEFT CLICK> (hand pointer turns black)
```

The name of the entity alone returns a batch of results (line 7) that include the official website for the Mini Cooper division of BMW and regional sites in USA, Canada and the UK, as well as four sponsored results. The user glances past the sponsored results (line 8) and very briefly at the first result (line 10). He then scrolls down the page (line 13) exposing the third result for Mini on Wikipedia, which also includes a thumbnail of an old Mini Cooper on the right side of the snippet. Immediately upon revealing this result, the participant inspects it (line 15) and clicks the link (lines 16–17) where he then easily finds the first production date (not shown). As in the previous case, this participant completes the task quickly in a mere 13.4 s. Again the name of the entity makes the task extremely easy to complete. The use of Wikipedia was also somewhat frequent in the data, being used in 18% of all cases. The participants used Wikipedia both to find factual answers to tasks, such as the first year the Mini Cooper was produced, as well as simply to find out what things are called (see next section).

Thus, somewhat like in conversation, the name of the entity is the most minimal and effective means of getting a search engine to recognize the intended referent. However, using the name requires prior knowledge on the part of the user. This is not always the case. In the following sections we see what users do when they *lack* prior knowledge of the entity's name.

Related Names

In two thirds of the cases in the data, users did not appear to know, or perhaps could not recall, the name of the entity in the task. Instead of producing the entity's name, they formulated their initial query either with a related name or a description. In this section we examine the former.

In 16% of all cases, users formulated their initial queries with a name that was in some way related to the entity in question. They thus display *some* prior knowledge of the entity although not its name:

- Plateau in *close encounters* (for *Devil's Tower*)
- Tokyo *cosplay* bridge (for *Jingu* Bridge)
- *Twilight* actor (for *Robert Pattinson*)
- German *chancellor* (for *Angela Merkel*)

Using related names is an effective strategy. In the following excerpt (3) the participant performs the Devil's Tower task (above) and displays knowledge, not of the place name, but of a related one. His initial query, "plateau in close encounters" (line 1), substitutes the proper name of the rock formation with a more generic descriptor. But it also includes a name reference to *Close Encounters of the Third Kind*, a movie in which Devil's Tower is prominently featured. It is this related name that enables the participant to complete the task.

```
(3) [10-Place2]
001 //plateau in close encounters - Yahoo! Search Results//
002 (0.4) {{looking where results should be}}
003 (results appear)
004 (0.2) {{Res#1 "CLOSE ENCOUNTERS"}}
005 <~MOUSE, SW>
006 (0.6) {{Res#1 "CLOSE ENCOUNTERS"}}
007 (0.5) {{Res#2 "Microsoft Word - Document19"}}
008 (0.5) {{Res#3 "Close Encounters of the Third Kind"}}
009 (0.1) {{glances near mouse pointer}}
010 <.MOUSE, above Res#4 "Features">
011 (0.4) {{URL of Res#3 "Close Encounters…"}}
012 (0.3) {{fi"hovering" in Res#4 "Features"}}
013 (0.7) {{URL of Res#3 "Close Encounters…"}}
014 (0.8) {{Res#4 "Features"}}
```

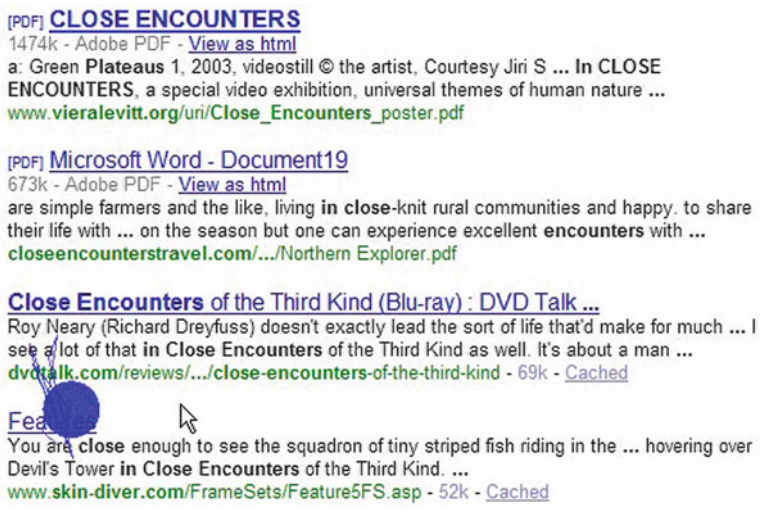

Fig. 10.4 Missing the entity name, "Devil's Tower" (excerpt 3, line 14)

```
015 <SCROLL DOWN 0.2>
016 (0.2) {{Res#6 "UFOArea: UFO And Alien"}}
017 (0.5) {{Res#5 "The Right Blue: Tales of Whales…"}}
018 (0.6) {{Res#6 "UFOArea: UFO And Alien Enc…"}}
019 <SCROLL DOWN 0.2>
020 (0.3) {{fiRes#8 "Globetrotter Games -Cosmic"}}
021 (0.4) {{fithumbnail of Res#9 "Topography - Pey"}}
022 (0.6) {{Res#9 "Topography - Peyote"}}
023 <SCROLL UP 0.3>
024 (0.2) {{fi"encounters" in search box}}
025 (0.2) {{"plateau in" in search box}}
026 <~MOUSE, N>
027 (1.1) {{on "plateau in" in search box}}
028 <.MOUSE, search box "pla|teau in close encounters">
029 <LEFT CLICK>
030 ((proceeds to replace "plateau" with "mountain"))
```

Unlike those in the previous excerpts, this user's initial query elicits a heterogeneous batch of search results: a movie poster, a travel brochure, a movie review, a skin diving site, accounts of encounters with Humpback whales. The user scans the first three results (lines 4, 6–8) before resting the mouse pointer near the fourth result but not on it (line 10). He continues scanning the URL of result 3 (line 11 and 13) and scans result 4 (lines 12 and 14, Fig. 10.4) before scrolling down the page (line 15). Result 4 is titled "Features," and its snippet begins with a line about getting close to "tiny striped fish" and its URL begins with "www.skin-diver.com." It thus appears to be entirely unrelated to the movie or the place for which the user is

searching, and indeed he quickly passes it over (line 16). However, result 4 actually contains that for which he is looking. In the second line of the snippet it says, "Devil's Tower in Close Encounters of the Third Kind," using a scene in the movie as an analogy, but the user does not fixate on the place name and appears to miss it. Instead he continues scanning results 5, 6, 8 and 9 (lines 16–22) before repairing his query (lines 24–30).

This participant continues on to replace the term "plateau" in his query with "mountain" (line 30), and click the result for "Close Encounters of the Third Kind" on Wikipedia. On this wiki page he finds the name "Devil's Tower" and follows the link to its own wiki page. With the place name confirmed, he then produces a second query repair but one that is very different. He enters "devil's tower wyoming motels" in the search engine box. In other words, he formulates a query that is very much like the one in excerpt 1, the entity name plus a task-relevant term. It takes him 83 s to find the name "Devil's Tower Wyoming" but then only 21.3 s to complete the task once he has the name. He thus conducts a *two-stage search:* a name search followed by a task-oriented search.

Similarly in the following excerpt (4), the same user employs the same strategy in completing the task to "Find the train station nearest to this bridge" accompanied by a photo of teens dressed in frilly dresses and other costumes on the Jingu bridge in Tokyo's Harajuku district. Instead of using the term "jingu bridge," the user composes the query: "tokyo cosplay bridge." He thus displays knowledge of the fact that the girls in the photo are participating in "cosplay," or "costume play." It is his knowledge of this related name that enables him to complete the task quickly.[1]

```
(4) [10-Place4]
005 //tokyo cosplay bridge - Yahoo! Search Results//
006 (0.5)
007 (search results appear)
008 (0.1)
009 <~MOUSE, W>
010 (2.1)
011 <.MOUSE, Res#1 "Harajuku - Wikipedia, the free encyc">
012 (0.5)
013 [<LEFT CLICK> (outline appears around result title)
014 [<~MOUSE, S>
015 (0.6)
016 //Harajuku - Wikipedia, the free encyclopedia//
017 (0.2)
018 <.MOUSE, to right of Res#1 "Harajuku - Wikipedia">
019 (0.1)
020 (page content appears)
021 (0.5)
022 <~MOUSE, W>
```

[1] Tracker loses eyes for most of this excerpt.

```
023 (0.6)
024 <.MOUSE, middle of second paragraph>
025 (1.8)
026 <SCROLL DOWN 0.2>
027 (0.3) {{photo caption "Girls at Harajuku Station…"}}
028 (0.3) {{photo of girls in dresses on Jingu bridge}}
029 (1.3) ((eye tracks disappears again))
030 (0.2) {{photo of rockabilly guys in Yoyogi Park}}
031 <~MOUSE, SE>
032 (0.4) {{photo of rockabilly guys in Yoyogi Park}}
033 (0.1)
034 <.MOUSE, caption of photo "Girls at Harajuku Station…">
```

The query, "tokyo cosplay bridge" elicits several highly relevant results (lines 5–7): the Wikipedia entry for "Harajuku," a YouTube video called "Harajuku Cosplay Bridge," a "Picture of Japanese cosplay girl posing on Jingu Bridge," and more. Although the eye-tracker loses the user's eyes for this portion, we can see that he takes only 2.2 s (lines 8–10), enough time to scan at least 4 results, to rest the mouse over result #1 (line 11). The participant then navigates to the Wikipedia entry (lines 13–20).

On the Wikipedia entry for "Harajuku," the participant likely scans the paragraphs above the fold (lines 21–25) before scrolling down (line 26). Scrolling down reveals a photo of the Jingu bridge with girls wearing dresses similar to those in the task image. The user glances at the caption of the photo (line 27), "Girls at Harajuku Station on a Sunday afternoon," as well as at the photo itself (line 28) before the eye track disappears again. He completes the task by placing the mouse pointer over the answer, "Harajuku Station" (line 34). The related name, "cosplay," the social activity for which Jingu bridge is perhaps most famous, enables this participant to complete the task in only 25.9 s.

We see then that if users have *some* knowledge of the entity in question and can produce a name of something related to it, they can also complete the tasks relatively quickly, especially with the aid of Wikipedia. Furthermore, in excerpt 3 the participant conducts a two-stage search. In stage 1 he searches for the name of the entity, and in stage 2 he uses the name in the query to complete the task. We will see more examples of two-stage searches in the next section.

Descriptions

While we have seen that both entity names and related names are very useful for completing tasks, in some cases users know nothing about the entity in question. However, this does not preclude them from completing the tasks. In almost two thirds of the cases in which users displayed no prior knowledge of the entities in the tasks, they nonetheless completed those tasks successfully. In these cases, users formulated their references to the entities as generic *descriptions* rather than as

names. These descriptions included details visible in the task images as well as terms in the task statement itself:

- Giant rippled rock (for *Devil's Tower*)
- Triangular building manhattan (for *Flatiron building*)
- Castle on a hill tourist place sea (for *Mont St. Michel*)
- Old personal computer (for *PARC Alto*)

In excerpt 5 the user must find the train station nearest to the Jingu bridge. This participant formulates her initial query as "tokyo bridge kid hangout" (lines 1), thereby exhibiting no prior knowledge of the name of the bridge ("Jingu" or "Harajuku") nor the name of the activity in which the kids are engaged ("cosplay"), although she gets the name of the city right ("tokyo"). Yet this mostly generic description is nonetheless effective in generating relevant search results.

```
(5) [12-Place4]
001 <TYPES "tokyo bridge kid hangout", 4.3>
002 (0.7) {{"searching…" in drop-down box}}
003 <ENTER> ("Connecting…" appears in page tab)
004 (0.2) {{fixes under search box}}
005 //tokyo bridge kid hangout - Yahoo! Search Results//
006 (0.4) {{fixes under search box}}
007 (search results appear)
008 (0.6) {{fixes under search box}}
009 (0.1) {{looks at Res#1 "YouTube - Harajuku Bridge"}}
010 <~MOUSE, S>
011 (1.2) {{Res#1 "YouTube - Harajuku Bridge"}}
012 <.MOUSE, right of Res#1 "YouTube - Harajuku Bridge">
013 (0.8) {{Res#1 "YouTube - Harajuku Bridge"}}
014 <~MOUSE, W>
015 (1.2) {{Res#2 "Harajuku Bridge"}}
016 <.MOUSE, Res#2 "Harajuku Bridge">
017 <LEFT CLICK> (hand pointer turns black)
018 [("Connecting…" appears in page tab)
019 [<~MOUSE, E>
020 (0.7) {{Res#2 "Harajuku Bridge"}}
021 //Harajuku Bridge//
022 (0.2) {{Res#2 "Harajuku Bridge"}}
023 <.MOUSE, right side of Res#2 "Harajuku Bridge">
024 (0.4) {{glances up at page tab "Harajuku Bridge"}}
025 (0.5) {{glances at search box and fixes on "bridge"}}
026 (page content appears)
027 (0.4) {{first sentence under "Harajuku Bridge"}}
028 <~MOUSE, SW>
029 (0.9) {{first sentence under "Harajuku Bridge"}}
030 (0.2) {{"Harajuku Station in Tokyo, Japan."}}
```

The initial query is successful at eliciting relevant results. Four of the six visible results mention “Harajuku Bridge” or “Jingu bashi.” The user spends 2.1 s inspecting the first result, “YouTube – Harajuku Bridge Jesus Freak” (lines 9, 11 and 13) and resting the mouse near it (line 12). She then continues on to the second result, “Harajuku Bridge” (at virtualjapan.com) inspecting it for 1.2 s (line 15), before opening the link (lines 16–17). On the result page (line 21), the user scans the first sentence (lines 27 and 29), which gives a brief description of the bridge, before fixing on the answer to the task, “Harajuku Station” (line 30). Although this user completes the task in 61 s using a brief description, she continues on to confirm her answer by searching for “Harajuku Bridge” on Flickr and finding a photo similar to the task image (not shown).

In 94% of the cases in which their initial queries consist of generic descriptions, users utilize some kind of image search at least once in the session to find or confirm a name, usually the “Images” tab of a search engine. When they employ this *image-searching* strategy, they attempt a *two-stage search*. In stage 1 they use generic descriptors combined with image search to try to find the name of the entity. In stage 2 they use the name in their query to complete the task. This second stage resembles cases in which users already know the name of the entity, for example:

Stage 1: Tourist attractions france
Stage 2: *Mont saint michel* review

Stage 1: First pc
Stage 2: *Xerox alto* computer history

Stage 1: Triangle building
Stage 2: *Flat iron* building

In the following excerpt (6), the user formulates his initial query as a description, “triangle building” (line 1); however, it fails to elicit relevant results.

```
(6) [14-Place1]
001 //triangle building - Yahoo! Search Results//
002 (0.4) {{glances at query in search box on page}}
003 (0.1) {{glances down to position of 1st result}}
004 (search results appear)
005 (0.9) {{skims Res#1 "Triangle Building Products"}}
006   (0.1)   {{glances   at   thumb   for   Res#5   "Triangle
Shirtwaist"}}
007 <~MOUSE, S>
008 (0.5) {{thumbnail for Res#5 "Triangle Shirtwaist"}}
009 (0.2) {{Res#4 "Triangle Building Networks"}}
010 (0.2) {{Res#5 "Triangle Shirtwaist Factory fire"}}
011 (0.1) {{glances up at Images tab}}
012 (0.2) {{glances between Web and Images tabs}}
013 (0.2) {{fi"triangle" in main search box}}
```

```
014 (0.2) {{looks at Web tab}}
015 (0.4) {{Images tab}}
016 <.MOUSE, "Images" tab>
017 <LEFT CLICK> (hand pointer turns black)
018 (0.2) {{Images tab}}
019 (0.1) {{glances at page tab}}
020 <~MOUSE, E>
021 (0.2) {{page tab}}
022 (0.2) {{Res#1 "Triangle Building Products"}}
023 //Yahoo! Image Search Results for triangle building//
024 (0.2) {{Res#1 "Triangle Building Products"}}
025 (image search thumbnails appear)
026 (0.3) {{img#1,2 "655016348 VFEYyXL jpg"}}
027 (0.2) {{img#1,4 "retaliat gif"}}
028 (0.6) {{img#1,1 "e8ff329a 7a974…99 jpg"}}
029 <SCROLL DOWN>
030 [<SCROLL DOWN 0.3>
031 [{{img#2,1 "20090625 143146…ew jpg"}}
032 (0.8) {{img#2,3 "transamerica pyramid"}}
033 (0.5) {{caption img#2,3 "transamerica pyramid"}}
034 (0.3) {{img#2,4 "government building…kremlin"}}
035 (0.2) {{fixes img#2,5 "i am locutus of borg"}}
036 (0.3) {{img#3,4 "illinois state capitol dome"}}
037 (0.4) {{img#3,3 "typical appartment block in…"}}
038 <SCROLL DOWN 0.3>
039 (0.2) {{img#4,4 "springfield the phanto…"}}
040 (0.3) {{on img#4,5 "austin texas state capitol"}}
041 (0.2) {{img#4,2 "illinois state capitol dome…"}}
042 <SCROLL UP 0.3>
043 (1.1) {{page tab "Yahoo! Image Search Results"}}
```

When the results appear (line 4), the user skims the first result (line 5), but it is a result for "Triangle Building Products," not for a famous triangular building. The user glances down at a thumbnail of a building included with result 5 (lines 6 and 8), but it is not the building from the task image. He glances at results 4 and 5 for only 0.2 s each (lines 9–10) before giving up on this batch of results. He then visibly changes strategies. He glances at the "Images" tab at the top of the search results page (line 15) and clicks it (lines 16–17). Filtering the results for images returns a page with ten visible thumbnails of buildings and triangles (lines 23 and 25). The user then begins to scan the thumbnails for the building in the task image (Fig. 10.5). In inspecting these image results, the user does not fixate on every one. Instead he skips around. While we might take this as evidence that he misses several of the thumbnails, cognitive research on "covert attention" suggests that he can still check images in the periphery of his visual field at some level of awareness (Wright and Ward 2008). The user scans each row in turn but skips around horizontally within them.

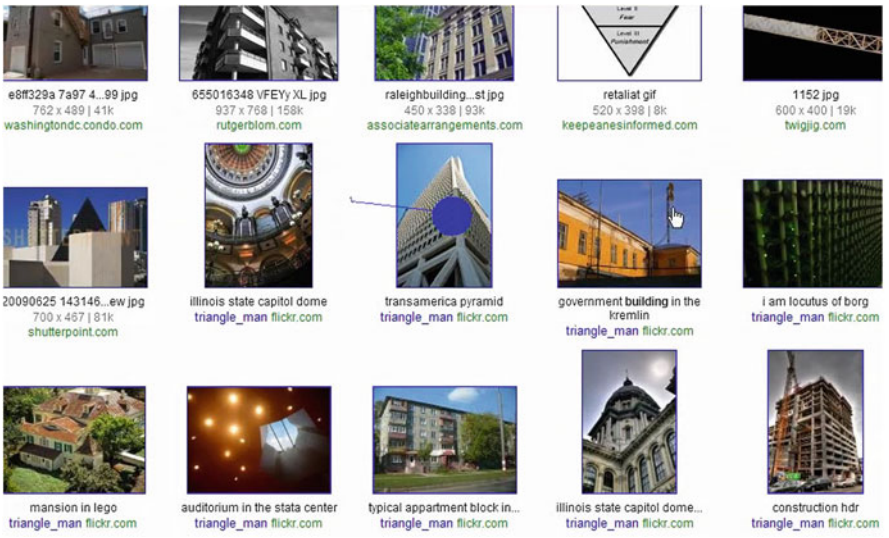

Fig. 10.5 Image-matching, *blue dot* marks eye gaze (excerpt 6, line 32) via Yahoo.com

First, he briefly touches the second image in the first row, an apartment building, with his gaze (line 26) followed by the fourth image in that row, a triangle chart (line 27). He then spends 0.6 s on the first image in the first row, a large house (line 28). Next the user scrolls down (lines 29–30) and continues scanning thumbnails in the second row: the first (line 31), the third (lines 32–33), the fourth (line 34) and the fifth (line 35). He continues scanning rows three (lines 36–37) and four (lines 39–41) in this fashion. Finding no matches, the user returns to the top of the page (lines 42–43) and repairs his query (not shown).

In this case the user conducts additional image searches for "chicago triangle building" followed by "nyc triangle building." The latter produces a match on Flickr: a photo of the same building in the task image. Unfortunately the caption reads, "One of the famous buildings you can find in New York." However, in the comments, another user asked, "OK which famous building?" and as our user scrolls down, he finds a response from a third person, "It is the Flat Iron building." This enables him to complete Stage 1 of his search, the name search. For Stage 2 he immediately submits a name query, "flat iron building" and proceeds to complete the task easily. He spends 2.1 min reviewing 81 thumbnail results to find the name "flat iron building," and then spends a mere 12.3 s to complete the task once he has the name.

While combining descriptive queries with image search was often a successful strategy, it failed one third of the time. For example in excerpt 7, the user is attempting to "Find a tourist review of this place," where the task image is that of St. Michael's Tower atop Glastonbury Tor in southwestern England. This user displays a lack of knowledge of this place by producing a descriptive initial query with no specific names, "green meadow one tower famous" (line 1).

```
(7) [15-Place3]
001 //green meadow one tower famous - Google Search//
002 (0.3) {{looks where results will appear}}
003 (search results appear)
004 (0.3) {{glances to first search result}}
005 (0.7) {{Res#1 "Singapore Condo Directory"}}
006 (0.2) {{glances up at "Images" tab}}
007 <~MOUSE, NW>
008 (0.5) {{fi"Images" tab}}
009 <.MOUSE, "Images" tab>
010 (0.2) {{fi"Images" tab}}
011 <LEFT CLICK> (outline appears around "Images" link)
012 (0.3) {{glances to where results will appear}}
013 (images results appear sans thumbnails)
014 (0.1) {{fixed in same place over first thumbnail frame}}
015 <~MOUSE, SE>
016 (0.3) {{fixed in same place over first thumbnail frame}}
017 (image result thumbnails appear)
018 (1.0) {{fiimg#1,1 "lush green meadows"}}
019 (0.2) {{glances across row}}
020 (0.3) {{img#1,3 "Maley Green"}}
021 (0.5) {{img#1,4 "of endless green"}}
022 (0.3) {{img#2,4 "About Meadows @"}}
023 (0.2) {{img#2,3 "The most famous"}}
024 <.MOUSE, margin right side of first row>
025 (0.4) {{img#2,1 "The green meadow"}}
026 (0.1) {{glances at img#2,2 "Towered over by"}}
027 <SCROLL DOWN 0.3>
028 (0.3) {{img#3,2 "BUY CLOCK TOWER 3D"}}
029 (0.6) {{img#3,3 "to see these green"}}
030 <SCROLL DOWN 0.2>
031 (0.4) {{img#4,2 "lush green meadows"}}
032 (0.3) {{img#5,2 "One of the many"}}
033 (0.4) {{img#5,3 "We come to a green"}}
034 (0.1) {{glances up at row 4}}
035 (0.4) {{img#4,4 "Grazing in fresh green"}}
036 <SCROLL UP 0.2>
```

When the search results appear (line 3), the user inspects the first result titled "Singapore Condo Directory – GREEN MEADOWS TOWER Information" (line 5), which appears to have no relation to the stone tower on the grassy hill. She immediately abandons the text results by switching to the image results (lines 6–11). When the thumbnail results appear (line 17), she begins scanning the images in the first row (lines 18–21) and working down through the second (lines 22–26), third (lines 28–29), fourth (line 31) and fifth (lines 32–33) rows. None of the thumbnails looks anything like the task image. The user then scrolls back to the top of the page

(line 36) and repairs her query (not shown). Overall this user formulates 14 variations of her descriptive query, the longest of which is "stone tower on a hill surrounded by green meadow tourist place england." She scans 289 thumbnail search results total and gives up after 7.5 min of mostly image searching.

Generic descriptions tended to consist of long queries of five words or more. Such queries have been found to perform poorly compared to shorter ones (Balasubramanian et al. 2010; Brendsky et al. 2008), as indeed they did in this study. While initial descriptive queries tended to be longer than name queries, descriptive queries often grew longer toward the middle of users' sessions if they could not find the entity. They might be characterized as "kitchen-sink queries" into which users progressively dump additional descriptive words and phrases until they become unwieldy. Like Aula et al. (2010) also found, the longest queries occurred in the middle of users' search sessions.

However, kitchen-sink queries sometimes get searchers on the right path, even though they are clumsy and may require the inspection of hundreds of thumbnails. In the following excerpt (8), the user must find a website from which he can buy a Dr. Martens Iconic Mary Jane shoe, which is indicated by a photo of this particular shoe. He begins with a description in his initial query "shoe thick sole strap" combined with image search. This excerpt begins 10.5 min into the session. The user has formulated 12 queries prior to this one and has scanned numerous thumbnails. For the current query, "woman casual walking shoe boot strap outdoor sole" (line 101) he has already inspected five pages of image results.

```
(8) [11-Product6]
101 //woman casual walking shoe boot strap outdoor sole//
102 (0.1)
103 (next image results appear)
104 (0.4) {{img#1,2 "MEN'S LEATHER"}}
105 (0.1) {{looks at img#1,3 "Offered by Super"}}
106 (0.1) {{glances across to img#2,2}}
107 (0.5) {{fiimg#2,2 "Frye Shoes Dolly"}}
108 <SCROLL DOWN 0.2>
109 (0.2) {{fiimg#3,2 "Women's"}}
110 (0.3) {{fiimg#3,1 "P.W. Minor Velvet -"}}
111 <SCROLL DOWN 0.2>
112 (0.2) {{fiimg#3,1 "P.W. Minor Velvet -"}}
113 (0.4) {{fiimg#4,1 "Teva Westwater"}}
114 <SCROLL DOWN 0.3>
115 (0.9) {{fiimg#5,4 "Dr Martens"}}
116 (0.3) {{fititle for img#5.4 "Dr Martens"}}
117 <~MOUSE, NE>
118 (0.2) {{fititle for img#5,4 "Dr Martens"}}
119 (0.4) {{fiimg#5,4 "Dr Martens"}}
120 (1.2) {{fititle for img#5,4 "Dr Martens"}}
121 <.MOUSE, center of img#5,4 "Dr Martens">
122 (0.3) {{fititle for img#5,4 "Dr Martens"}}
```

```
123 (0.3) {{fiurl for img#5,4 "altrec.com"}}
124 (0.2) {{fiimg#5,4 "Dr Martens"}}
125 (0.1) {{fisize of img#5,4 "Dr Martens"}}
126 <~MOUSE, S>
127 (0.9) {{fisize for img#5,4 "altrec.com"}}
128 <.MOUSE, url img#5,4 "altrec|.com">
129 (0.4) {{fisize of img#5,4 "altrec.com"}}
130 (0.1) {{fiimg#5,4 "Dr Martens"}}
131 <~MOUSE, N>
132 (0.4) {{fiimg#5,4 "Dr Martens"}}
133 <.MOUSE, center of img#5,4 "Dr Martens">
134 <LEFT CLICK> (hand pointer turns black)
135 (0.2) {{fiimg#5,4 "Dr Martens"}}
136 //Google Image Result for http://www.altrec.com/…//
```

The user inspects the rows of thumbnails from top to bottom, starting with the first row (lines 104–105), then second row (lines 106–107), third row (lines 109–112) and fourth row (line 113). Upon revealing the fifth row (line 114), he quickly glances at the fourth and last image (line 115). He has encountered this thumbnail before in two prior searches and has even clicked on it twice. The shoe in the image is indeed a Dr. Martens brand shoe, but it is *not* the Iconic Mary Jane for which he is looking. Yet it appears to be the closest match that he has found so far. The user inspects the whole result carefully. He fixates on the thumbnail for almost a second (line 115), fixates on the image title, "Dr. Martens," for 0.5 s (lines 116 and 118), as he begins to move the pointer (line 117) toward the result. He looks at the thumbnail again (line 119) before fixating on the image title for 1.5 s (lines 120 and 122) and parking the mouse pointer in the center of the thumbnail (line 121). He continues inspecting the result by looking at its URL (line 123) and its file size (lines 125, 127 and 129). While doing so, he moves the mouse pointer over the URL, "altrec.com," below the image (lines 126 and 128). He then looks back up at the thumbnail (lines 130, 132 and 135) while moving the pointer back to the center of it (lines 131 and 133) and clicks (line 134, Fig. 10.6).

On the result page (line 136), this user examines two names: *altrec.com* and *Dr. Martens.* The former is the name of the current website, whose logo is prominently displayed in the upper left corner. The user submits the query, "altrec woman shoe casual strap" and clicks the images tab (not shown). However, the search engine returns no image results. The user then replaces the name "altrec" with "dr. martens" in the query and resubmits. But again the search engine fails to return image results, so he ends up clicking the first sponsored result, "Dr. Martens Shoes - Cheap." Finally this gets him on the right path, a shoe store website filtered for Dr. Martens shoes. He thus partially completes the name search with "dr. martens woman shoe casual strap," but because the name is not specific enough, he does not find the shoe immediately. He continues scanning six pages of image results on the shoe store website itself before finding the right one. After 13.4 min, during which time he scans 1,246 thumbnail images, the user finally completes the task. Indeed for this user, knowing the name "Dr. Martens Iconic Mary Jane" at the start would

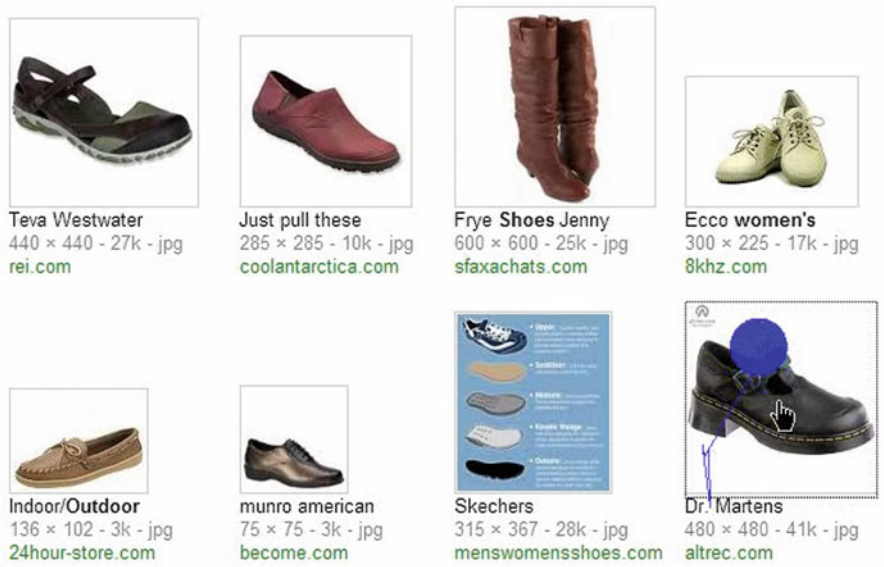

Fig. 10.6 Inspecting a similar shoe (excerpt 8, line 134) via Yahoo.com

have been worth a 1,000 pictures. While inefficient compared to names, generic descriptions nonetheless enable users to find pages even when they have no prior knowledge of a thing.

When Names Fail

While we have seen that users may formulate their queries as names or descriptions, we can see further that there is an ordered relationship between these two strategies. In other words, a name if available should be tried first. We see in the following cases that users try a name in their initial query but then discover a problem with it. They then switch to using descriptions. In no cases in our data do users start with a description and then when it fails switch to the name.

- *Ozzie osbourne* → aging balding frizzy hair metal rocker
- *Half dome yosemite* → famous gray monolith rock
- *Chateau d'if castle* → french castle walkable island

For example in excerpt 9, the user is trying to "Find a recent news story about this rock star" and is presented with a photo of Ronnie James Dio. As her initial query, she uses an incorrect name, "ozzie osbourne" (line 1).

```
(9) [12-Person2]
001 //ozzie osbourne - Google Search// ("Images" tab)
002 (0.1)
```

```
003 (0.2) {{glances at query in main search box}}
004 (results and thumbnails appear)
005 (0.3) {{fiimg#1,1 "Ozzy Osbourne was"}}
006 (0.3) {{fiimg#1,2 "Ozzie Osbourne"}}
007 (0.2) {[fiimg#1,3 "Ozzy Osbourne"}}
008 (0.2) {{inactive page tab "Person 2"}}
009 <~MOUSE, W>
010 (0.5) {{fiinactive page tab}}
011 <.MOUSE, inactive page tab>
012 [<LEFT CLICK> (pointer turns black)
013 [{{glances at face of person in task image}}
014 (0.6) {{face of person in task image}}
015 (0.1) {{glances at inactive page tab "ozzie osbourne}}
016 <~MOUSE, E>
017 (0.4) {{fiinactive page tab "ozzie osbourne"}}
018 <.MOUSE, inactive page tab "ozzie osbourne">
019 (0.1) {{glances down at task image}}
020 [<LEFT CLICK> (pointer turns black)
021 [(page content for tab "ozzie osbourne" appears)
022 (0.3) {{fiimg#1,3 "Ozzy Osbourne"}}
023 (0.3) {{fiimg#1,4 "Ozzy gets so"}}
024 (0.5) {{img#2,3 "Ozzy Osbourne, the"}}
025 (0.3) {{fiimg#2,4 "Still the fucking Prince"}}
026 <~MOUSE, SE>
027 (0.3) {{fiimg#2,1 "Ozzy Osbourne Bio"}}
028 (0.4) {{fiinactive page tab "Person 2"}}
029 <.MOUSE, page tab "Person 2">
030 <LEFT CLICK> (pointer turns black)
```

Immediately after the image results appear, the user very briefly glances at the first three thumbnails of the first row (lines 5–7). She then goes back to the other page tab, which contains the task instructions and image (lines 8–12). She briefly inspects the task image for less than a second (lines 13–14), before returning to the image results (lines 15–21). She glances at the third thumbnail in the first row again (line 22) and glances at four more thumbnails (lines 23–27) before abandoning this batch of results. In other words, she checks these thumbnails against the task image and quickly sees that they do not match.

After the name "ozzie osbourne" fails as a query, the user tries another possibly related name, the band "judas priest" (not shown), which similarly fails to elicit any image matches. At this point the user changes strategies by formulating her query as a description instead of a name. She submits the query, "aging balding frizzy hair metal" and proceeds to formulate seven additional descriptive queries. In the end she spends 4.7 min mostly scanning 132 thumbnail images before giving up on the task. But what is most notable about this case is the order in which she employs the different references forms: first a specific entity name, then a possibly related name and then a generic description.

Discussion

The preceding analysis reveals how users deal with a particular kind of trouble, and likely "frustration," when interacting with Internet search engines. In querying search engines, users must somehow *refer* to the entities for which they seek relevant web pages. When they already know the name of the entity in question, participants tried the name first, and when it was correct, the Internet search engines delivered relevant results almost instantly. However, when participants lacked the name, or tried a name that failed, they fell back on alternative forms of reference in an effort to find the name. If they possessed *enough* prior knowledge of the entity to produce a related name, such as "close encounters" for the place featured in the movie *Close Encounters of the Third Kind*, participants could fairly easily find the entity name, especially through Wikipedia. However, when participants displayed *no* prior knowledge of the entity, they employed a different strategy to find the name: they formulated queries using generic descriptors, such as "giant rippled rock," to generate a set of image results and then scanned the image thumbnails for the entity in an effort to find its name. Once located through this procedure, participants used the entity name in the same kinds of ways that participants who knew the names from the start did. Such practices can thus be characterized as *two-stage searches*, the first stage of which consists of a search-for-the-name and the second stage of a completion of the task using the entity name. The organization and goal of each stage are thus considerably different. The study thus demonstrates a preference for naming in web search, somewhat similar to the preference for naming in human conversation (Sacks and Schegloff 1979; Moore 2008). In both cases, people appear to try a name if they can, but fall back on various forms of *description* if the name fails.

Focusing on what users do when they lack the name of the thing for which they are searching throws some affordances of two Internet resources into relief. First, we see that the *image search* feature of major search engines is useful not just for finding images, but also for finding *names*. When conducting the name search part of a two-stage search, the user is *not* interested in the resulting images *as* images but rather as means for locating an entity. In the context of such an activity, the quality and size of the image results do not matter as much as the pairing of the image result with the entity's name, which does not always occur. Second, Wikipedia is also very useful for finding entity names. In 18% of the cases, for example excerpts 3 and 4, users exploited Wikipedia. Not only does Wikipedia offer basic information on virtually any known entity, it also tends to pair an image of the entity in question with its name.

This study also contributes to the literature on query formulation. Like prior studies (Balasubramanian et al. 2010; Brendsky et al. 2008), it finds that verbose queries perform poorly compared to shorter "keyword" queries; however, it adds a new insight to this observation: it is not just the length of a query that matters. The long, "kitchen-sink" queries observed in this study were not only ineffective because they are verbose but more importantly because they lacked a critical type of "keyword,"

the entity name. Thus, not all keywords are equal. Names make especially effective keywords whether they are the name of entity in question or they are simply closely related to it. In addition, this study identifies a particular pattern in query reformulation – searching for an entity name using images followed by using the name in a subsequent query – which is not mentioned in the current query reformulation literature (Rieh and Xie 2006; Liu and Belkin 2008).

This study further demonstrates the situated utility of the text-based approach to *image search* (Chai et al. 2007) in the context of web search. Although we observed that generic descriptions of the entity images, and especially "kitchen-sink" queries, indeed performed poorly compared to names in terms of efficiency, they were used deliberately in the absence of those names and for the purpose of finding those names. In other words *name searches* appear to be used as a kind of workaround for managing a particular type of knowledge and interaction problem. A name may be worth a 1,000 pictures, but a two-stage search can still get the job done under less than ideal circumstances.

In this study, the participants exhibited a preference for minimization in choosing a form of reference somewhat similar to that observed in conversation (Sacks and Schegloff 1979; Moore 2008). Entity names were used first if possible, but longer descriptive references were used when names were unknown or problematic. Therefore, when search queries are verbose *due to their descriptive nature*, they can be taken as signs of interactional trouble and of a knowledge gap on the part of the user. In other words, the occurrence of verbose queries suggests that the user chose to relax the preference for minimization for some reason. We see in this study that, that reason may be that they lack the entity name, but there may be other reasons as well. For example, Aula et al. (2010) also find that long queries tended to be used during difficult tasks, especially toward the middle of sessions, but due to different kinds of troubles. The wording of their tasks included the relevant entity names for the participants but were difficult because the information sought was scarce on the Internet. After a few failed queries, those participants often formulated their queries as grammatical questions thereby increasing query length. Therefore future studies are necessary to determine when verbose queries indicate a knowledge problem or an information scarcity problem or possibly other kinds of problems as well.

Future research should also examine additional forms of reference in search-engine interaction. One such form is image queries or Google's 'search by image' function. Using this method, the user submits an image file or URL *as* a query. It is therefore quite similar to *pointing* in face-to-face conversation: instead of naming or describing the thing, the speaker simply points and says "*that*." At the time of this study, 'search by image' was a fairly new feature of Google search and none of the participants used it, although they were instructed to use *any* website they wanted. Submitting an image of the entity in question, in the absence of using its name, could thus be another solution to the reference problem these participants faced. There are however a few limitations with this form of reference. First, it requires the user to possess an image of the entity in question. While they did in fact have an image in this study, in real life they may not, if for example they are trying to search for "the thing they saw on TV last night." Second, a very cursory test of Google's

'search by image' function shows that it works well for some images and entities but not others. While it easily found and named the image of Devil's Tower used in this study, which was borrowed from Wikipedia, it did less well with other images of Devil's Tower. Similarly it did very well with an image of a Francis Francis X5 espresso machine used in this study, which was borrowed from commercial websites, but it failed miserably with an image of the same espresso machine taken by the author. It thus appears that Google's 'search by image' does very well at finding the *same* image on the Internet but not necessarily with finding an entity with an image that is not already on the Internet. A follow-up study should examine how participants use 'search by image' with the same tasks in this study and with a variety of images.

Conclusion

We have seen a particular class of troubles that users may encounter in interaction with Internet search engines, as well as users' practices for managing these troubles. We have also seen evidence that suggests that current Internet search engines are highly name-oriented: they work well in recognizing names of entities but not other forms of reference.

This analysis of referential practice raises certain questions for search-engine design. Should search engines serve results the same way for users doing a name search (stage 1) as for users doing a proper search (stage 2)? Stage-1 troubles involve knowledge of the thing itself, whereas stage-2 troubles involve the availability of information about the thing online. Therefore each kind of trouble requires a different kind of solution. Stage-2 troubles may call for more sophisticated query techniques, whereas stage-1 troubles call for educating the user. The latter might be achieved by displaying a wider diversity of entities, more thumbnails or more links to Wikipedia on the search results page. In addition, such a name-search mode could highlight entity names on the search results page, treat search terms as descriptors rather than as proper names and provide an "I don't know what it's called" link for entering name-search mode. Future work should explore these questions.

Because lacking an entity name is a function of the particular searcher's current state of knowledge and the particular entity of interest, this kind of search trouble may be encountered by *any* kind of searcher, novice and expert alike. For any user, no matter how knowledgeable, there are always some sets of entities of which they know nothing. But while this study identifies a generic sort of practice, one limitation is that it does not provide an estimate of how prevalent such a trouble might be. Such an estimate would require naturally occurring data and a different method. However, from the preceding analysis, we know something about the structure of name searches: they tend to involve long queries (5+ terms), long search sessions (5–10 queries) and the browsing of image results. This basic pattern seems possible to detect automatically. Therefore, in a future study, the *frequency* of name searches could likely be estimated using quantitative analysis of server logs, using these three indicators. By enabling an understanding

of the formal structures of user practices, computer interaction analysis can also inform the design of measures in large-scale studies.

Finally, to put the current topic in its wider context, *referential practice* is a basic component of human-computer interaction, as well as of social interaction (Moore 2008). Earlier command-line interfaces required users to refer to entities entirely by *naming* them; graphical user interfaces (GUI) enabled users to *point* to entities (if they are present) for the first time; and search enables users to refer to entities by *describing* them, as well as naming them or pointing to them using images. Thus, future research should examine how these three resources for referring – naming, pointing and describing – get used alone or sequentially in interactions with other kinds of computer applications. It should also demonstrate how such referential practices are related to the particular circumstances of the setting, such as the availability of the entities of interest, the knowledge of the users *and* of the system, and the input methods (e.g., command line, pointer, search) available.

Transcript Notation

`(0.7)` – Gaps between system events timed in tenths of seconds.
`{{skims 1st sentence}}` – Double braces indicate descriptions of eye-gaze behavior during a given time period.
`(spinning icon appears)` – Single parentheses indicate computer display events.
`<LEFT CLICK>` – Angled brackets indicate input actions.
`<TYPES "yahoo", 0.2>` – Text input. Underlining indicates characters entered and number indicates duration of typing.
`<~MOUSE, NW>` – Start of mouse movement, with approximate compass direction.
`<.MOUSE, "location">` – Mouse stop, which may include description of location or quoted text. Underlining indicates precise mouse position.
`[`
`[` – Open brackets indicate lines overlap in time.
`//window title or URL//` – Double slashes indicate window title.

Acknowledgments I thank Prasad Kantamneni of Yahoo!'s Human Perception Center Of Excellence for the use of eye-tracking facilities and help with study design.

References

Aula A, Jhaveri N, Käki M (2008) Information search and re-access strategies of experienced web users. In: Proceedings of the WWW2005, ACM Press, New York, pp 583–592

Aula A, Khan RM, Guan Z (2010) How does search behavior change as search becomes more difficult? In: Proceedings of the CHI2010, ACM Press, New York, pp 35–44

Balasubramanian N, Kumaran G, Carvalho VR (2010) Exploring reductions for long web queries. In: Proceedings of the SIGIR'10, ACM, New York, pp 571–578

Brendsky M, Croft WB (2008) Discovering key concepts in verbose queries. In: Proceedings of the SIGIR'08, ACM, New York, pp 491–498
Chai JY, Zhang C, Jin R (2007) An empirical investigation of user term feedback in text-based targeted image search. ACM Trans Inform Syst 25(1):1–25
Fallows D (2008) Search engine use. Pew Internet & American Life Project. www.pewinternet.org.
Feild H, Allan J, Jones R (2010) Predicting searcher frustration. Proceedings of the SIGIR'10, ACM Press, New York, pp 34–41
Garfinkel H (1967) Studies in ethnomethodology. Prentice-Hall, Englewood
Hassan A, Jones R, Kinkner KL (2010) Beyond DCG: user behavior as a predictor of a successful search. In: Proceedings of the WSDM'10, ACM, New York
Hitwise (2010) Top 20 Sites and Engines as of 9/20/10. www.hitwise.com
Jefferson G (2004) Glossary of transcript symbols with an introduction. In: Lerner G (ed) Conversation analysis: studies from the first generation. John Benjamins, Amsterdam, pp 13–31
Liu Y, Belkin NJ (2008) Query reformulation, search performance, and term suggestion devices in question-answering tasks. In: Proceedings of the IIiX'08, information interaction in context, ACM, London
Moore RJ (2008) When names fail: referential practice in face-to-face service encounters. Lang Soc 37(3):385–413
Moore RJ, Churchill EF (2011) Computer interaction analysis: toward an empirical approach to understanding user practice and eye gaze in GUI-based interaction. Comput Support Cooper Work 20:497–528
Moore RJ, Churchill EF, Kantamneni RGP (2011) Three sequential positions of query repair in interactions with internet search engines. In: Proceedings of the CSCW'11, ACM, New York
Rieh SY, Xie H (2006) Analysis of multiple query reformulations on the web: the interactive information retrieval context. Inform Process Manage 42:751–768
Sacks H, Schegloff EA (1979) Two preferences in the organization of reference to persons in conversation and their interaction. In: Psathas G (ed) Everyday language: studies in ethnomethodology. Irvington, New York, pp 15–21
Sacks H, Schegloff EA, Jefferson G (1974) A simplest systematics for the organization of turn-taking for conversation. Language 50:696–735
Suchman L (1987) plans and situated actions: the problem of human-machine communication. Cambridge University Press, New York
Wright RD, Ward LM (2008) Orienting of attention. Oxford University Press, New York

Part IV
Future Trends in Mobile Speech

Chapter 11
Summarizing Opinion-Related Information for Mobile Devices

Giuseppe Di Fabbrizio, Amanda J. Stent, and Robert Gaizauskas

Abstract Reviews about products and services are abundantly available online. However, gathering information relevant to shoppers involves a significant amount of time reading reviews and weeding out extraneous information. While recent work in multi-document summarization has attempted to some degree to address this challenge, many questions about extracting and aggregating opinions remain unanswered. This chapter demonstrates a novel approach to review summarization, using three techniques: (1) graphical summarization; (2) review summarization; and (3) a hybrid approach, which combines abstractive and extractive summarization methods, to extract relevant opinions and relative ratings from text documents. All three methods allow a consistent approach to preserve the overall opinion distribution that is expressed in the original reviews.

G. Di Fabbrizio (✉)
Lead Member of Technical Staff, AT&T Labs – Research,
180 Park Avenue – Building 103, Florham Park, NJ, USA
e-mail: pino@research.att.com

A.J. Stent
Principal Member of Technical Staff, AT&T Labs – Research,
180 Park Avenue – Building 103, Florham Park, NJ 07932, USA
e-mail: stent@research.att.com

R. Gaizauskas
Department of Computer Science, University of Sheffield, Regent Court,
211 Portobello Street, Sheffield, S1 4DP, UK
e-mail: R.Gaizauskas@sheffield.ac.uk

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_11,
© Springer Science+Business Media New York 2013

Introduction

The last 5 years have seen transformational advances in the use of advanced networked mobile devices, enabling users to merge their online and offline lives as never before. One of the daily life tasks that is most illustrative of this transformation is *purchasing*. Consumers on-the-go increasingly rely on internet search to find services and products, and on online reviews to select from among them. The actual purchase of the selected service or product may take place online or at a *bricks-and-mortar* location. A study conducted by The E-tailing Group[1] describes an emerging breed of shopper, the *social researcher*, who seeks out opinions expressed by online peers before making buying decisions. According to this research, 78 % of the 1,200 sampled consumers spent more than 10 min reading reviews online. Additionally, 65 % meet the definition of social researchers and 86 % of social researchers rated online reviews and product ratings an *extremely* or *very important* factor influencing their buying decisions. Another study[2] carried out by comScore and The Kelsey Group revealed that a significant portion of offline product and service sales (24 % of the 2,000 interviewed users) are made after consulting online reviews while three quarters of consumers who consulted online reviews reported that the reviews had a significant influence on their purchase. Retailers and service providers are also recognizing that there is a growing crowd of shoppers who rely on reviews to learn about products and services and, ultimately, make decisions about spending their money (Chevalier and Mayzlin 2006; Duan et al. 2008; Park et al. 2007).

Reviews about products and services are abundantly available online. However, even from traditional PC interfaces, identifying relevant information in product and service search results involves a significant amount of time reading reviews and weeding out extraneous information. The mobile device presents new challenges and opportunities to developers of both search and review browsing services:

- Screen size – Mobile devices have relatively small displays and limited navigation capabilities. Requested information should be presented in only one to two screens to minimize vertical and horizontal scrolling. This presents opportunities for intelligent pruning, grouping, and summarization of search results and reviews.
- Time – Mobile users are often *on-the-move* with limited time to refine search criteria or select relevant information from a long list of results. Since time is of the essence, the presented information should be targeted to the user's specific goal.
- Location – Mobile users are highly focused on executing geographically *local plans* such as finding restaurants, entertainment events, or retail stores. The precision

[1] www.marketingcharts.com/direct/social-shopping-study-defines-new-breed-of-shopper-the-social-researcher-2347.

[2] www.comscore.com/press/release.asp?press=1928.

Fig. 11.1 Spek4it: local business search by voice

of presented information can be improved by considering the user's location and nearby businesses. For example, a system may choose to only present search results within walking or driving distance of the user.

- Personalization – For mobile users, personal data (e.g., search and purchasing histories) can be used to improve the precision of search results and the informativeness of reviews.

Although constrained by the same factors, typically, mobile search and mobile review browsing are treated as different tasks using a combination of poorly integrated algorithms. This leads to inefficiencies and decreases user satisfaction.

For example, imagine that a consumer wants to buy *Skechers* shoes. The consumer would first use a local mobile search engine to find nearby shoe stores (see Fig. 11.1). The search engine might re-rank search results by using geographic information about the current user's location (Stent et al. 2009) – or an explicitly requested location – and, optionally, re-score the final results based on domain knowledge and/or the user's search history. Once in the store, the user may use a separate internet search to find and browse online reviews and ratings for particular types of shoes. *Opinion mining* and *sentiment analysis* methods can be applied to extract the targets, and the relative polarity (e.g., positive, negative, or neutral), of the opinions expressed in the reviews (Hu and Liu 2004; McDonald et al. 2007; Pang and Lee 2008). Lastly, the user must synthesize (or *summarize*) all the facts, opinions, and ratings read in the previous step to find the most desirable option.

While there exist relatively accurate and efficient methods for the two steps in this process (search, and sentiment analysis), *summarization* of evaluative text (e.g., documents containing opinion or sentiment-laden text) is a fairly new

technology. Yet it is important for mobile users, due to the small amount of screen real estate they have access to and the distractions of other tasks they may be performing.

In this chapter, we describe three examples of product and service review summarization techniques. The first approach is a graphical summarization method that predicts the "number of stars" (the polarity and strength of opinion) a reviewer may have assigned to a specific topic based on the opinion expressed in the textual review. In this scenario, a set of predefined topics are ranked on a scale from one (poor) to five (excellent) stars and graphically visualized on a mobile device. The second method describes a review summarization technique based on natural language generation where review ratings and other review features are used to automatically generate a short natural language description of the opinions expressed across the reviews. The last proposed technique is a hybrid approach that combines abstractive and extractive summarization methods and interleaves quotes from reviewers directly into the natural language summary.

The rest of the chapter is structured as follows: section "What Is in a Review?" describes the main characteristics of product and service reviews. Section "Case Study in Mobile Search and Reviewing: Have2eat" presents a case study in mobile searching and reviewing. section "Sentiment Analysis Using Multi-rating Multi-aspect Predictions" illustrates sentiment analysis methods to predict review ratings. section "Review Summarization" shows text summarization approaches to review synthesis. section "Conclusions" concludes and presents future work.

What Is in a Review?

What makes reviews different from other user-generated content such as blogs, wikis, social network documents, or message boards? Reviews are usually either about a single *product*, e.g., consumer goods including digital cameras, DVD players, or books; or related to a *service* like lodging in an hotel or dining in a restaurant. Typically a product or service has several ratable *aspects* (sometimes referred also as *topics* or *features*). This means that a review can be viewed as a set of aspects, each with an associated *rating*. Ratings define the strength and polarity of opinions and typically range over integer values; they often visualized with star symbols.

YELP,[3] for example, combines local business reviews and social networking capabilities to rank businesses. Figure 11.2 shows the page dedicated to the Japanese restaurant *Santouka Ramen* located in Los Angeles, CA. The business has been reviewed by 944 customers, each of whom wrote a textual descriptions of their experience – the *review* – and rated their **overall** experience with a star-based score – the overall *rating* – from one (poor) to five (excellent).

[3] www.yelp.com.

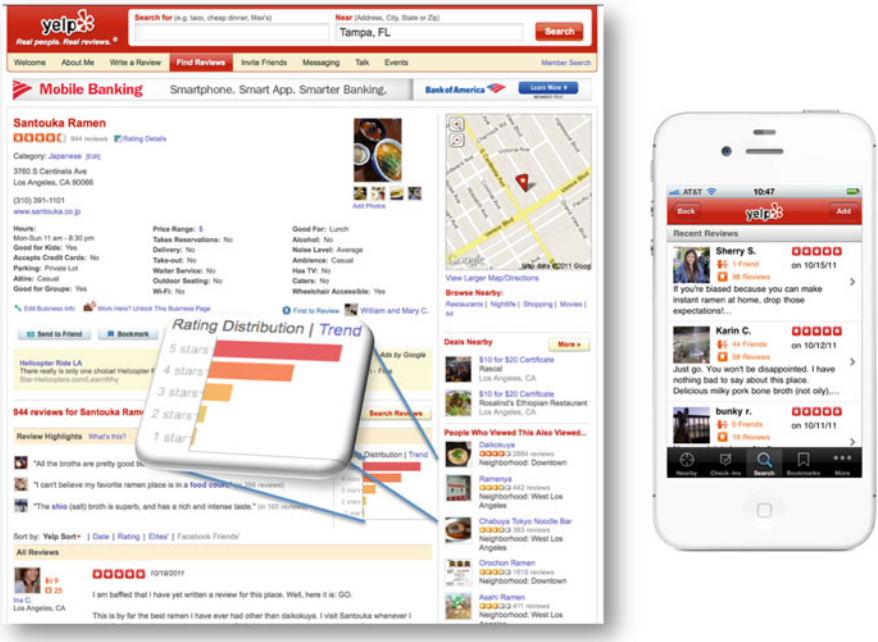

Fig. 11.2 Yelp's Web browser- and mobile-based visualization of reviews for the *Santouka Ramen* restaurant in Los Angeles, CA

The rating distribution graph visualized in the *"Reviews Highlights"* section of Fig. 11.2 presents a half-bell shaped curve indicting that most reviewers' opinions about this business are biased towards the positive end of the ratings. But a better understanding of the opinions expressed by reviewers would require decomposing the overall ratings based on aspects that are relevant to a dining experience. Each contributing aspect would be associated with a more detailed rating and with relevant parts of the reviewer's textual description. This would allow a reader to easily identify whether users most appreciated the good service, the quality of the special of the day or the décor. With only overall ratings, finding these fine-grained details requires skimming through most of the 944 reviews, some of them very lengthy and with substantial amounts of extraneous or irrelevant information. Thus, the abundance of online products and service reviews does not optimally help users, who must search and skim through thousands of documents to identify the information relevant to them.

These considerations suggest that there is a substantial need for automatic methods to summarize the content of reviews. This is particularly true for mobile users, who are constrained by small displays. However, traditional *extractive* document and multi-document summarization techniques, where relevant fragments of text are selected from input documents and concatenated into a consistent summary

(Goldstein et al. 2000), are not helpful for evaluative texts covering multiple aspects of an entity, with a range of opinions for each aspect. Without a robust semantic model of entities, their aspects, and opinion strengths and polarities, automatic extractive summarizers produce incoherent summaries. In other words, automatic summarization of evaluative texts requires two components: sentiment analysis and extraction, and then summarization.

As pointed out by Wiebe et al. (2004), evaluative language presents specific linguistic characteristics that are usually missed in traditional natural language processing approaches. *Sentiment analysis* techniques for evaluative texts must identify the linguistic elements realizing sentiment, their target domain-relevant aspects, and their semantic orientation in the context of the document. These elements are often *lexical*, ranging from single words (e.g., *fantastic, dreadful, good*) to more complex syntax structures (e.g., *"to put it mildly", "stand in awe"*, *"the most disappointing place that I have stayed"*). Wiebe et al. (2004) refer to three types of sentiment clues: (1) *hapax legomena* – unique words appearing only once in the text; (2) *collocations* – word ngrams frequently occurring in subjective sentences; and (3) *adjectives* and *verbs* – extracted by clustering according to a *distribution similarity* criterion (Lin 1998). Additionally, the contexts where the clues appear in the sentences play a key role in determining actual polarity of the opinions being expressed. Polanyi and Zaenen (2005) describe additional clues – contextual value shifters – that modify the positive or negative contributions of other clues. Consider the positive word *efficient*; when modified by the intensifier *rather*, the resulting *rather efficient* is a less strongly positive expression.

One final consideration for review analysis and summarization is authenticity. Fake, or *spammy* reviews are increasingly common. Detecting fake reviews is very hard; recent contributions (Feng et al. 2012) show some promising techniques to find deceptive product reviews by analyzing sudden changes in the rating distribution footprints. The likely presence of fake reviews in any dataset makes it even more important that review summaries accurately capture the full range of opinions expressed.

Case Study in Mobile Search and Reviewing: Have2eat

Have2eat[4] is a popular restaurant search and reviewing application available for iPhone and Android-based devices. During search, Have2eat uses geo-location information (from the GPS device or explicitly entered by the user) to produce a list of matching restaurants sorted by distance and located within a specific radius from the originating location. During browsing of search results, when restaurant reviews

[4] www.have2eat.com.

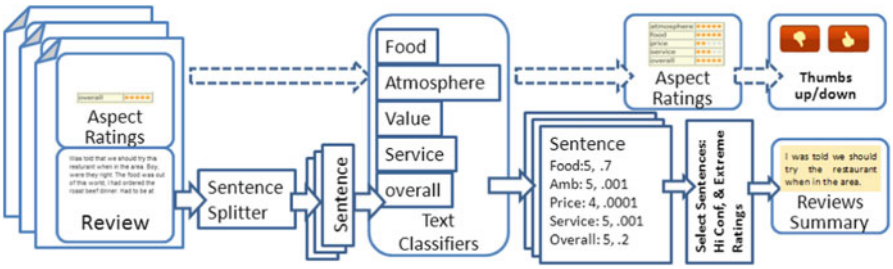

Fig. 11.3 Graphical and textual summarization process

are available, a compact one-screen digest displays a summary of the reviews posted on the web by other customers. Customers can expand to read a full review page and also enter their own ratings, comments and feedback. The review summaries are visualized on the mobile screen:

- **Graphically** by *thumbs-up* (positive reviews) and *thumbs-down* (negative reviews) for different aspects of the restaurant;
- **Textually** by a few sentences selected from review texts that best summarize the opinions about various aspects of the restaurant expressed in the reviews.

There are similar mobile applications obtainable either on the Apple iPhone App Store or as web-based mobile applications, such as Zagat,[5] UrbanSpoon,[6] YP Mobile,[7] and Yelp,[8] but, to the extent of our knowledge, most of them are only focused on the restaurant search task. When available, restaurant reviews are simply visualized as a contiguous list of text snippets with the overall experience rating. None of the listed applications include extended rating prediction or review summarization.

System Description

The Have2eat system architecture is composed of three parts: (1) a geographic restaurant search engine based on YellowPages listings; (2) a multi-rating multi-aspect (MRMA) predictive model for sentiment analysis – illustrated in Fig. 11.7 and described in section "Graphical Summarization by Thumbs Up/Down", and (3) graphical and textual summarization – shown in Fig. 11.3 and described in section "Textual Summaries by Sentence Selection".

[5] mobile.zagat.com.

[6] www.urbanspoon.com.

[7] m.yp.com.

[8] m.yelp.com.

Table 11.1 Mapping example between ratings and thumbs up/down. Ratings of 3 are considered neutral and ignored in this mapping

	Reviews			Thumbs	
	a	b	c	Up (%)	Down
Atmosphere	3	2	4	50	50 %
Food	4	4	5	100	0
Value	3	2	4	50	50 %
Service	5	5	5	100	0
Overall	4	4	5	100	0

Graphical Summarization by Thumbs Up/Down

We used the MRMA model described in detail in section "Sentiment Analysis Using Multi-rating Multi-aspect Predictions" to predict star-ratings associated with domain-specific attributes from restaurant reviews. To save room on the small mobile screen, the predicted ratings were mapped onto *thumbs-up* or *thumbs-down*. Table 11.1 shows an example of this mapping.

Textual Summaries by Sentence Selection

Figure 11.3 shows how summary sentences are selected from textual reviews. As described in section "Sentiment Analysis Using Multi-rating Multi-aspect Predictions", we trained predictive models for each aspect of the restaurant. To select summary sentences we split the review text into sentences.[9] Using the predictive models and iterating over the restaurant listings, for each sentence in each review we get five aspect-specific ratings and confidence scores for those ratings. We then select a few sentences that have extreme ratings and high confidence and present them as summary text.

We evaluated these summaries using the following metrics.

1. **Aspect accuracy**: How well selected sentences represent the aspect they are supposed to.
2. **Coverage**: How many of the aspects present in the textual reviews are represented in the selected sentences.
3. **Rating consistency**: How consistent the selected sentences with the summarizing aspect ratings are.
4. **Summary quality**: Manual evaluation – how good the summaries are based on subjective human judgments; automatic evaluation – how good the summaries are based on ROUGE (Lin 2004), an automatic multi-document summarization score that compares summaries to manually created reference summaries (*gold* standard).

[9] For this purpose we used a statistical sentence splitter trained on email data and using n-gram and word part-of-speech features.

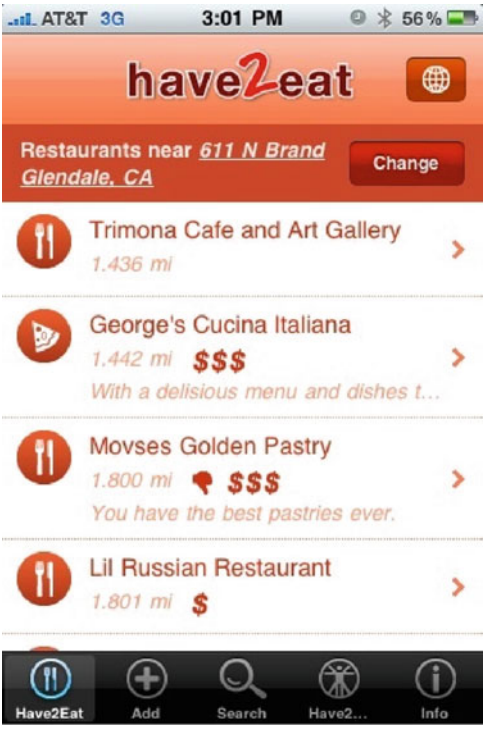

Fig. 11.4 Have2eat listings screen shot on iPhone

How It Works

When launching the application, users are presented with a list of 20 nearby restaurants. The user can browse more restaurants by tapping on a link at the bottom of the page. For each listing we show the distance from the user's current location and, if available, we provide an overall thumbs-up or thumbs-down, price information and the summary sentence with the highest confidence score across aspects. Figure 11.4 shows an example of the *List* page. If users want a list of restaurants for a different location they can tap the *Change* button at the top of the page. This action will bring up the *Location* page where the user can enter city and state and/or a street address.

Users can select a restaurant in the list to view details, see Fig. 11.5. Details include address, phone number and thumbs up/down for each of the overall, food, service, value and atmosphere aspects. The user can provide feedback (a new review) by tapping on the thumbs-up or thumbs-down buttons, as well as by leaving a comment at the bottom of the screen. This page also includes a few summary sentences with extreme ratings and high confidence scores. An example of selected sentences with their polarity is shown in Table 11.2. By tapping on any of the sentences the users can view the full text of the review from which the sentence was selected.

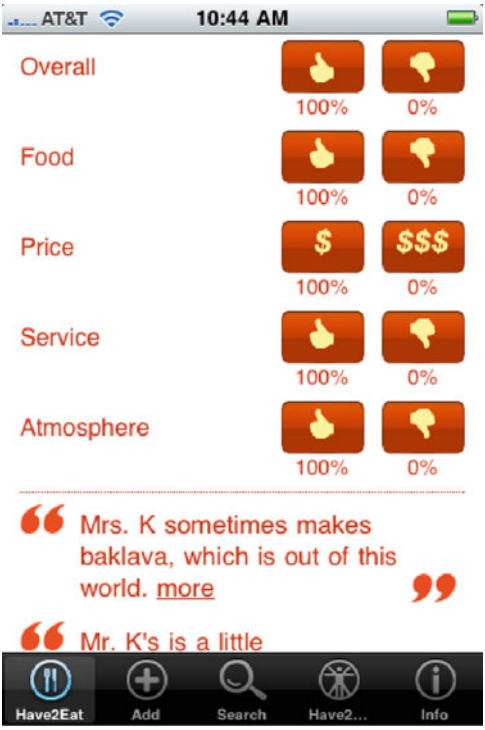

Fig. 11.5 Have2eat automatically predicted aspect ratings and text summary

Table 11.2 Example of extracted summaries

Restaurant 1 (3 reviews)	
+	The soups are GREAT! Everything that we have ever ordered has exceeded the ex…
+	Delivery is prompt and credit cards are welcome
+	Their chicken fried rice is the second best in Southern California
Restaurant 2 (8 reviews)	
+	Great tasting burgers, friendly fast service!
+	The inside is warm and even though the chairs looked uncomfortable, they were not at all
–	Too many other places to try to worry about getting mediocre food as a high price
Restaurant 3 (4 reviews)	
+	The salads are tasty, the breadsticks are to die for
–	We waited approximate 10 more minutes and then asked how much longer
+	A fun place to go with family or a date
–	If you like salt then this is the place to go, almost everything is full of s…

Users can also add a new restaurant by tapping the *Add* icon in the tab bar.

Figure 11.6 shows a Review page, displaying the review selected in the *Details* page along with any other reviews that exist for the restaurant. Users can give feedback on whether or not they found the review helpful. Users can also add a review by tapping on a link at the bottom of the page.

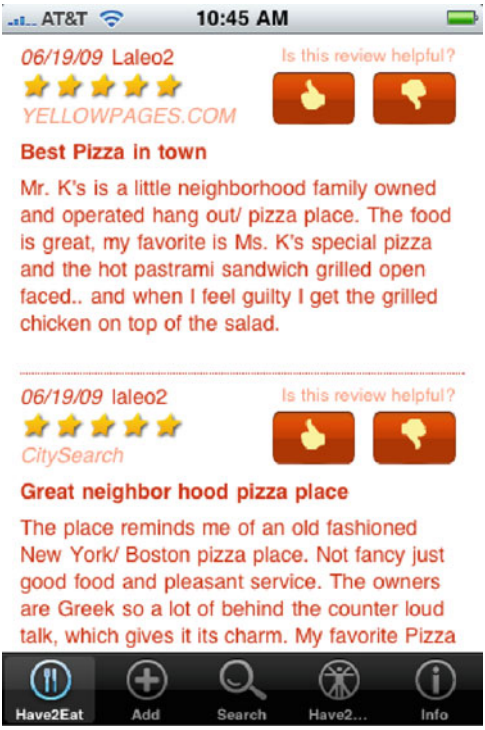

Fig. 11.6 Have2eat review details

Now that we have seen how sentiment analysis and review synthesis can be used by mobile users, we turn to an examination of the technology that enables these functionalities. In section "Sentiment Analysis Using Multi-rating Multi-aspect Predictions", we present the sentiment analysis system used in Have2eat, and in section "Review Summarization" we present approaches to review summarization, including the one used in Have2eat.

Sentiment Analysis Using Multi-rating Multi-aspect Predictions

As mentioned before, textual reviews are abundant, and most online reviews are already tagged by reviewers with an overall star rating that should capture the general opinion expressed in the textual review about the evaluated entity. However, a single overall rating does not provide enough information about how the aspects of the reviewed entity contribute towards the overall rating.[10]

[10] Some review web sites try to obtain aspect-specific ratings, but this is a longer and more difficult process for reviewers.

For example, in the case of restaurant reviews, reviewers usually focus on one or more of the following five aspects: *food*, *atmosphere*, *value*, *service* and *overall experience*. Given the rating information associated with the textual review, is it possible to learn from textual data how to automatically predict the ratings for each aspect? In other terms, can we model the way the reviewers map a textual description into a number of stars? These questions refer to what has generally been defined as *multi-rating multi-aspect* (MRMA) modeling, but often appears with different labels such as *multi-aspect ranking* (Snyder and Barzilay 2007), *rating inference* (Shimada and Endo 2008), *multi-facet ranking* (Baccianella et al. 2009), *rated aspect summarization* (Lu et al. 2009), and *multi-aspect rating prediction* (Lu et al. 2011). Applying MRMA predictions to a set of textual reviews would provide a synthetic representation of the opinions expressed in the reviews that can be easily visualized on mobile small screens.

Previous Work

Most previous work on MRMA for sentiment analysis uses relatively simple features such as word ngrams, contrastive word-pairs, and part of speech patterns. Baccianella et al.'s work on MRMA for the hotel review domain (Baccianella et al. 2009) focuses on feature engineering – finding the best features for this task. Starting from a simple bag-of-words baseline, they first add part-of-speech (POS) tags patterns such as *"ADJ NN"* or *"NN VB ADJ"*, and then include normalized phrases based on the General Inquirer (Stone et al. 1966) lexicon, so that the expressions like *"horrible location"* or *"disgusting location"* are both replaced with *"[negative] location"*. Best performance is obtained by combining all the features.

In addition, most previous work on MRMA for sentiment analysis models this task as a classification problem (where the predicted output is an element of an unordered set of classes) (Dave et al. 2003; Pang and Lee 2004; Shimada and Endo 2008), while the rating prediction task is more accurately modeled as an **ordinal regression** problem (where the predicted output is an element of an *ordered* list of classes). Baccianella et al. (2009) characterize MRMA as an ordinal regression task, but they actually conduct their experiment using support vector regression (SVR) modeling, where the predicted output is a real number. Machine learning approaches for ordinal regression, such as the Perceptron Rank algorithm (Crammer and Singer 2001), have been used for MRMA (e.g. Lu et al. 2011; Snyder and Barzilay 2007; Titov and McDonald 2008), but may perform worse than approaches like SVR because they are very sensitive to parameterization and data ordering.

Most previous work on MRMA for sentiment analysis has incorrectly treated all aspects to be rated as independent of each other. In possibly the most complete work on this task to date, (Snyder and Barzilay 2007) use MRMA modeling for the restaurant domain assuming inter-dependencies among aspect ratings. They capture the relationships between the ratings via an *agreement relation* that describes the likelihood that a user will express the same rating for all the rated aspects. They

show that modeling these relationships helps to reduce the *rank loss*, the average difference between the ratings predicted by their model and the true rating given by the reviewer.

Snyder and Barzilay's approach uses two types of model: aspect-specific rating prediction models and a *meta-model*, or agreement model, that takes into account relationships between aspects. Both types of model are trained from a corpus of labeled reviews, using features including unigram and bigrams appearing in the corpus with a frequency greater than three, and features that might capture highly contrastive ratings, such as the presence of word-pairs like *"delicious"* and *"dirty"*. The ratings predicted by the aspect-specific models are compared by the meta-model, which may adjust them to minimize the overall disagreement (*grief*) between related aspects.

In the next section, we present a detailed set of experiments evaluating MRMA for sentiment analysis in the restaurant review domain. We extend the feature space proposed in previous work, and compare several classification and regression algorithms.

Exploring New Approaches to MRMA

As discussed previously, while MRMA is most accurately described as an ordinal regression task, the most mature and highly accurate machine learning methods are those for multi-class classification. We therefore use and compare several different approaches to aspect rating prediction for MRMA: multi-class classification, numeric regression, and ordinal regression.

In our experiments, each reviewed entity can have a number of domain-specific aspects (e.g., for the restaurant review domain, the aspects we consider are *food quality*, *service*, *atmosphere*, *value* and *overall experience*). Each aspect can have a rating value from 1 (poor) to 5 (excellent). With **multi-class classification**, we use one class for each rating value, and one classifier for each aspect. In this approach, the ordering inherent in the rating values is not directly used to help classification. Multi-class classification algorithms include traditional large margin classifiers such as binary *one-vs-all maximum entropy* (MaxEnt) classifiers (Berger et al. 1996), *support vector machines* (SVMs) with different kernel flavors (Haffner et al. 2003), and *boosting* (Schapire and Singer 2000).

With **numeric regression**, we use one classifier for each aspect, but the output is a real-number prediction which is then mapped into discrete rating intervals. Numeric regression implementations include statistical regression in a general linear model, and neural networks trained using the back-propagation algorithm.

Ordinal regression is the most theoretically suitable approach to rating prediction. We use one classifier per aspect, and *multiple thresholds* ($r-1$ thresholds are used to split r ranks). The perceptron-based ranking algorithm *PRank* (Crammer and Singer 2001) is one possible implementation of ordinal regression, but we present other options in the next sections.

Table 11.3 Restaurant review corpus statistics

Restaurants	3,866
Reviewers	4,660
Reviews	6,823
Average reviews per restaurant	1.76
Number of sentences	58,031
Average sentences per review	8.51

Table 11.4 Restaurant review ratings distribution per aspect (percentages and absolute number of ratings)

Ratings	1 *(poor)*	2 *(below average)*	3 *(average)*	4 *(above average)*	5 *(excellent)*	Total *(absolute no. of ratings)*
Atmosphere	6.96	7.81	14.36	23.70	47.18	6,761
Food	8.24	6.72	9.86	18.53	56.65	6,769
Service	11.83	6.12	11.91	22.00	48.14	6,788
Value	9.37	7.57	13.61	23.27	46.18	6,761
Overall	10.48	8.19	10.17	20.47	50.69	6,800

Data

The experiments described in the next sections are based on reviews mined from a restaurant review web site around the end of 2008. Each review document comprises the user's textual review and the user-assigned aspect-specific ratings. We only considered reviews rated with all the following aspects: *food*, *service*, *atmosphere*, *value* and *overall experience*. Partially rated reviews were discarded. Actual restaurant corpus statistics (after removing reviews with only partial ratings) are reported in Tables 11.3 and 11.4.

At the time of mining, reviews were collected for about 3,800 restaurants. There was an average of about two reviews per restaurant, and around eight sentences per review. Table 11.4 shows review rating distribution for each aspect. Ratings are skewed toward the high end, with 70 % or more ratings of *above average* (rank 4) or *excellent* (rank 5).

Classification and Regression Modeling

To predict aspect ratings of restaurants from their textual reviews, we first trained classification models in the restaurant domain with different feature combinations, and then compared them with two traditional regression models. We evaluated our models using *rank loss* – the difference between the predicted rating for each aspect, and the true rating for each aspect given by the human reviewer. The lower the rank

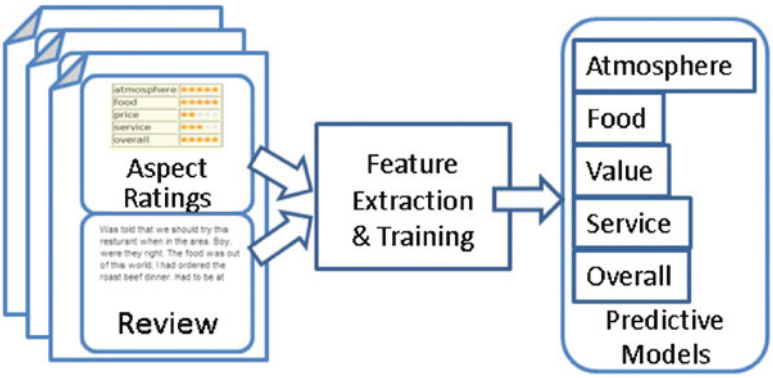

Fig. 11.7 MRMA predictive model training

loss, the more accurate the rating prediction. All our experimental results are averages of 10-fold cross-validation over the review examples: this means that we conducted each experiment 10 times, each time randomly dividing the data into 90 % training and 10 % testing data, and then we computed the average rank loss over all the test partitions. Figure 11.7 illustrates the training process.

All our experiments were performed using the AT&T machine learning toolkit LLAMA (Haffner et al. 2003, 2005) which is a highly efficient and scalable implementation of large margin classification algorithms, such as MaxEnt, SVMs, and AdaBoost, both for binary and multi-class classification.

In these experiments, we treated aspect rating predictors as independent of each other. For each aspect, predictor models were trained independently and were used independently to predict ratings.

Feature Selection

First, we performed feature engineering experiments to select good features for our later experiments. For our feature engineering experiments we used MaxEnt classifiers. We experimented with different combinations of features: like (Baccianella et al. 2009) we used word unigrams and bigrams occurring more than three times in the training data, but we also used *word chunks*, and *part-of-speech* (POS) chunks. We obtained chunks by extracting noun (NP), verb (VP) and adjective (ADJP) phrases from the reviews. We removed modals and auxiliary verbs from VPs, pronouns from NPs, and we broke chunks containing conjunctions. Table 11.6 shows examples of extracted word and part-of-speech chunks from reviews.

Bags-of-words (i.e., unigrams) capture basic word statistics, while bigrams permit modeling of very local word context. POS chunks and word chunks can provide a simple form of word sense disambiguation and, at the same time, aggregate co-occurring words (i.e., collocations), such as *sautéed_onions*, *buffalo_burger*,

Table 11.5 Average rank loss using MaxEnt classifiers with different feature sets

Aspects	Unigram	Bigram	Word chunks	Word chunks + POSchunks	Unigram + Overall rating
Atmosphere	0.740	0.763	0.789	0.783	0.527
Food	0.567	0.571	0.596	0.588	0.311
Value	0.703	0.725	0.751	0.743	0.406
Service	0.627	0.640	0.651	0.653	0.377
overall	0.548	0.559	0.577	0.583	–
Average	0.637	0.652	0.673	0.670	0.405

etc. We also hypothesized that by using word chunks we can keep information bearing phrases and remove the rest.

Most web-based reviews do not provide per-aspect ratings of products or services. However, they often give an overall rating evaluation. We therefore also experimented with using the overall rating as a feature.

The results of our experiments are shown in Table 11.5. As can be seen, in spite of the richness of word and POS chunks, the models using word unigrams perform better. We can attribute this to data sparsity. This result is in line with the findings of Pang et al. (2002). The last column of Table 11.5 shows that use of overall rating as an input feature significantly improves performance. This validates our intuition that per-aspect ratings are highly correlated with overall ratings.

We conducted additional experiments including different combinations (and permutations) of features such as unigrams, bigrams, and word chunks, but the overall performances were not better than the average performances of the features listed in Table 11.5. For the remaining experiments, we used only the unigram features. Since overall ratings given by reviewers may not always be available, we did not continue to use them as input features.

Classification Versus Regression Models

Second, we compared our MaxEnt classification-based approach to MRMA to: (1) a simple baseline (2) a numeric regression approach (*multilayer perceptron* (MLP) Sarle 1994) and (3) an ordinal regression approach (*perceptron-based ranking* (PRanking) Crammer and Singer 2001). For our baseline approach, the predicted rating for each aspect is simply the most frequently occurring rating in the training data, which is "5" for each aspect (see Table 11.4).

Table 11.7 shows the results of our experiments for the restaurant review domain. The second column shows the results for our baseline approach; the average baseline rank loss is greater than one. The third column shows the results from the numeric regression approach with MLP using *backpropagation* (Sarle 1994) training and uniformly distributed thresholds for class selection; this is the second best performing approach. The fourth column corresponds to PRanking, a classical

Table 11.6 Example review and extracted word chunks

Review sentences

```
<s>Poor service made the lunch unpleasant.</s>
<s>The staff was unapologetic about their mistakes they just didn't seem to
care.</s>
<s>For example the buffalo burger I ordered with sauteed onions and fries
initially was served without either.</s>
<s> The waitress said she'd bring out the onions but had I waited for them before
eating the burger the meat would have been cold.</s>
<s>Other examples of the poor service were that the waitress forgot to bring out
my soup when she brought out my friend's salad and we had to repeatedly ask to
get our water glasses refilled.</s>
<s> When asked how our meal was I did politely mention my dissatisfaction with
the service but the staff person's response was silence not even a simple I m
sorry.</s>
<s>I won't return. </s>
```

Word chunks

```
poor_service made lunch unpleasant
staff unapologetic mistakes n't care
example buffalo_burger ordered sauteed_onions fries served
waitress said bring onions waited eating burger meat cold
other_examples poor_service waitress_forgot bring soup brought friend salad
repeatedly ask_to_get water glasses_refilled
asked meal politely_mention dissatisfaction service staff_person response silence
not simple sorry
n't return
```

Part-of-speech chunks

```
NNP_NN VBD NN JJ
NN JJ NNS RB VB
NN NN_NN VBD NN_NNS NNS VBN
NN VBD VB NNS VBD VBG NN NN JJ
JJ_NNS JJ_NN NN_NN VB NN VBD NN NN RB VB_TO_VB NN VBZ_VBN
VBD NN RB_VB NN NN NN_NN NN NN RB JJ JJ
RB VB
```

Table 11.7 Average ranking loss using different predictive models

Aspects	Baseline	MLP with backpropagation	PRank	MaxEnt
Atmosphere	1.036	0.772	0.930	0.740 ± 0.022
Food	0.912	0.618	0.739	0.567 ± 0.033
Value	1.114	0.740	0.867	0.703 ± 0.028
Service	1.116	0.708	0.851	0.627 ± 0.033
Overall	1.077	0.602	0.756	0.548 ± 0.026
Average	1.053	0.694	0.833	0.637 ± 0.020

ordinal regression algorithms where the ranks are first mapped into real numbers and then optimal thresholds are determined iteratively to minimize the rank loss; this approach performs badly compared to numeric regression and multi-class classification. The MaxEnt (Berger et al. 1996; Haffner et al. 2005) classification results appear in the last column, where we also show the standard deviation over

the ten cross-validation trials; this is our best performing approach. This outcome is somewhat unexpected, since classification does not consider ordering among classes like regression does. However, binary and multiclass classification methods are the most investigated and refined machine learning algorithms, and this may account for the difference in performance. In the next section, we present a method for performing ordinal regression by performing binary classification, thus taking advantage of the most advanced machine learning algorithms while retaining the fidelity of ordinal regression for the MRMA task.

Reducing Ordinal Regression to Binary Classification

Machine learning toolkits like LLAMA (Haffner et al. 2003, 2005) have been continuously improved to optimize performance and scalability for classification tasks, but these advantages are not available for ordinal regression tasks, where the ordering information should help to obtain better prediction models. Moreover, simply transforming ordinal classes into numerical values and applying numeric regression may not produce the expected performance gain, since such a transformation assumes that the distance between classes is known and matches the numerical distance, which is not necessarily the case. However, there are techniques that can be used to transform a classification algorithm into an ordinal regression algorithm by simply changing the feature space and adapting the decoding process. Such techniques can leverage existing and well-tuned classification methods without requiring changing the core loss function and the regularization parameters.

Li and Lin (2007) present a *reduction* framework that extends binary classification to ordinal regression. This general method expands the training data by adding a *mislabeling cost* matrix. The classifier trained from the expanded training set can be extended as well to produce ordinal regression values. Li and Lin demonstrate that binary classifiers that generalize well could also support ordinal regression rules that generalize well, so that well-tuned classifiers will also preserve good performance when used for ordinal regression tasks.

In practical terms, the reduction method can be implemented with a simple encoding schema. With a K-class ordinal regression problem, for each training sample $\{\mathbf{x}\}^{(i)}$ and ordinal label $y \in \mathbb{Y} = \{1,2,\ldots,K\}$ we introduce a binary coding vector $\mathbf{E}^{(i)}$ of $(K-1)$ values $(e_1^{(i)},\ldots,e_{K-1}^{(i)})$ to form an extended feature set by appending $\mathbf{E}^{(i)}$ to $x^{(i)}$. The output label $y_k^{(i)}$, where $k \in (1,\ldots,K-1)$ is assigned to be either 0 or 1 based on the following function:

$$y_k^{(i)} = \begin{cases} 1 & \text{if } rank(y^{(i)}) > k \\ 0 & \text{otherwise} \end{cases}$$

where $rank(y^{(i)})$ returns the ordinal position of the output label. Table 11.8 shows an example of feature space expansion with $K=5$.

Table 11.8 Ordinal regression to binary classification reduction for $K=5$

	Output mapping					
Feature space	1	2	3	4	5	$Rank(y^{(i)})$
$(\mathbf{x}^{(i)},1,0,0,0)$	0	1	1	1	1	>0
$(\mathbf{x}^{(i)},0,1,0,0)$	0	0	1	1	1	>1
$(\mathbf{x}^{(i)},0,0,1,0)$	0	0	0	1	1	>2
$(\mathbf{x}^{(i)},0,0,0,1)$	0	0	0	0	1	>3

Table 11.9 Average rank loss using ordinal regression with feature reduction

Aspects	MaxEnt	Linear-SVM	Poly-SVM	Boosting
Atmosphere	0.708	0.698	0.687	0.739
Food	0.547	0.546	0.524	0.607
Value	0.625	0.605	0.595	0.672
Service	0.690	0.661	0.655	0.754
Overall	0.556	0.540	0.510	0.610
Average	0.625	0.610	0.594	0.676

The *feature space* column characterizes the expanded training sample for $\mathbf{x}^{(i)}$. For $K=5$, each sample is expanded into four instances and the new feature vector output values are determined accordingly to the *output mapping* columns. For instance, if the original output rank $rank(y^{(i)})=1$, all the output values are zero (first column), if the rank is 3, the first two training samples have an output value of 1 and the rest 0 (third column).

At classification time, first the input features are expanded as described before, then $(K-1)$ classifications are performed with the new jointly trained classification model. Finally, all the $(K-1)$ binary output values are considered at the same time and the actual ordinal output is decoded according to the *Output mapping* columns in Table 11.8 or, equivalently, by the following mapping function $\hat{y}^{(i)} = k+1$ where k identifies the first not null predicted value $\hat{y}_k^{(i)}$ when proceeding backwards from $\hat{y}_{k-1}^{(i)}$ to $\hat{y}_1^{(i)}$.

Table 11.9 shows the average rank loss using ordinal regression with feature reduction. We experimented with four large margin classification algorithms. The best performances are obtained with SVM models, both with linear and polynomial kernels. Considering the average rank loss across the aspects, the polynomial SVM ordinal regression method performs around 7 % absolute better than our MaxEnt classification method.

Review Summarization

The output of our MRMA-based sentiment analysis system consists of, for each sentence in each review, predicted ratings for each aspect, and an estimate of the system's confidence in each rating. This information can be used to produce a

summary review for an entity. Summary reviews are particularly useful for mobile users, as they can be read quickly on small-screen devices. In this section, we briefly look at three approaches to review summarization.

What Is in a Good Review Summary?

A review summary should *accurately*, *concisely* and *informatively* present the consensus opinions expressed in the input reviews, for all aspects of the reviewed entity that the user cares about. A review summary may fail to be accurate if it presents an opinion that is not in fact the consensus opinion; this can happen if the sentiment analysis system fails, or if there is no consensus opinion (for example, if sentiment is divided on an aspect, or if sentiment has changed over time). A review summary can also be inaccurate if the input reviews are anomalous in some way, for example if there are very few of them or if very few discuss one or more aspects. A review summary may fail to be concise if it is repetitive (for example, presents information about the décor repeatedly), or if it covers information the user does not want to know about, such as aspects the user does not care about, or extraneous information from an input review (e.g., the name or history of the reviewer). Finally, a review summary may fail to be informative if it does not include information the user does care about (even if that information is not in the input reviews), or if it is incoherent and hard to understand. Obviously, there can be a tradeoff between conciseness and informativeness; on a mobile device the desire for conciseness may be used to exclude information, while on a traditional large-screen device informativeness may win.

Review summaries, if consisting of text, should also be *readable* and *coherent*. If they are not, users may become frustrated or confused, or may not trust the content of the review summary.

In this section we will look briefly at three methods for review summarization: the first is to summarize the reviews graphically, while the other two produce textual review summaries. *Extractive summarization* involves the selection and knitting together of text fragments from the input reviews, while *abstractive summarization* involves the generation of new text sentences to express information about the range, polarity and strength of the opinions expressed in the input reviews.

Graphical Summaries

A graphical review summary presents, using icons, tables and/or charts, the opinions expressed in the input reviews. One simple example of a graphical review summary would be a Table indicating, for each aspect of an entity, the percentage of reviews expressing positive and negative opinions of that aspect (see the top part of Fig. 11.5). Graphical review summaries can be very concise and easy to read; however, they may fail to be accurate if they oversimplify the degree of

disagreement in the input reviews (for example by merging very and somewhat negative ratings, and very and somewhat positive ratings). They may also lack informativeness, because they cover up details present in the original reviews, particularly when an aspect was controversial (for example, a bimodal distribution for food quality, where all users who hated the food quality were vegetarians, and all users who liked it were meat eaters). Graphical summaries can also accompany extractive or abstractive summaries, as in Have2eat (see section "Graphical Summarization by Thumbs Up/Down").

Extractive Summaries

Text summarization is well-established area of research, and most approaches to the task are *extractive*, that is, they select and stitch together pieces of text from the input documents. However, most summarization techniques focus on distilling *factual* information by identifying the input documents' main topics, removing redundancies, and coherently ordering extracted phrases or sentences. Most contributions have also been developed using corpora with well-formed documents such as news stories (Conroy et al. 2005; McKeown et al. 2002), medical articles (Elhadad et al. 2005), biographies (Copeck et al. 2002), technical articles (Saggion and Lapal 2002), and blogs (Mithun and Kosseim 2009). Summarization of evaluative text such as reviews is substantially different from the traditional text summarization task: instead of presenting facts, the summarizer must present the range and mean of opinions, and instead of focusing on one topic, the summarizer must present information for multiple attributes of an entity. In addition, as observed in Ku et al. (2006), traditional summarization techniques discard redundancies, while for summarization of sentiment-laden text, similar opinions mentioned multiple times across documents are crucial indicators of the overall strength of the sentiments expressed by the writers.

Carenini et al. (Carenini and Cheung 2008; Carenini and Pauls 2006; Carenini et al. 2012) were the first to implement an extractive summarizer for evaluative text. Their summarizer, MEAD*, was adapted from the general-purpose MEAD open-source text summarizer (Radev et al. 2004). It takes as input the output of a sentiment analysis system like the one presented earlier in this chapter; specifically, a set of text reviews, each comprising a set of sentences, and each sentence labeled with a rating for each domain aspect. It assigns a single score to each sentence by summing up the aspect-specific ratings for the sentence. Then, it makes a "bin" for each aspect, and puts into each bin all sentences with a non-null rating (or a rating with high enough confidence) for that aspect. Finally, to construct the summary it iterates through the bins, selecting at each step the sentence with the highest sentence-level score and removing that sentence from all bins (to avoid selecting the same sentence multiple times), until the summary is of the desired length.

Di Fabbrizio et al. (2011) propose an alternative, graph-based method for extractive summarization over evaluative texts. Their system is called STARLET. First, a

directed acyclic graph is constructed as a series of layers. The first layer has j nodes representing the *j* sentences in the input reviews; the second layer has $j(j-1)$ nodes representing two-sentence summaries; the third layer has $j(j-1)(j-2)$ nodes representing three-sentence summaries, and so on. Each node is weighted using a scoring function that takes into account multiple features of the represented summary, such as number of aspects mentioned, their polarity, and the linguistic quality of the included sentences. A* search is used to find an optimal path through the graph. In an evaluation, STARLET performed better than MEAD and a simple baseline when evaluated using automatic metrics, and STARLET was rated by human users as producing summaries equivalent to those of MEAD in terms of grammaticality and better in terms of coverage.

Abstractive Summaries

Before there were mobile applications for local business search, there were dialog systems (e.g., Johnston et al. 2001; Liu et al. 2010. Dialog systems are subject to many of the same constraints as mobile applications; in particular, the communication channel is narrow (the conversation should be short/the device has a small screen), so there is a compelling need for summarization whenever information needs to be presented to the user. In purchasing, two types of information need to be presented to the user: lists of objective information about matching entities, and reviews of subjective information pertaining to selected entities. Considerable research has been done on the automatic generation of *recommendations* and *comparisons* for spoken dialog systems. These are similar to review summaries in that they produce evaluative texts expressing raters' opinions of aspects of one or more chosen entities. They are also different: they need not be based on input text reviews, but only on input ratings; and they involve the automatic production of new text. In addition, while the goal of an extractive summary is simply to compress the information in the input documents, natural language generation systems can produce summaries designed to serve other goals, such as recommendation, comparison, justification or exemplification.

Abstractive summaries can be almost as concise as graphical summaries, and are often more coherent and readable than extractive summaries. However, the systems for generating abstractive summaries involve more engineering than those for producing extractive summaries and, like graphical summaries, abstractive summaries can obscure detail present in input text reviews.

An abstractive summarizer is a natural language generation system, which takes as input a set of per-aspect ratings for one or more entities. The system produces text through a sequence of decisions. First are *what* decisions: deciding what entities to discuss, and which aspects to discuss for each entity. Then there are *how* decisions: deciding how to group and order information about entities and aspects into paragraphs and sentences, and what words to use to communicate the polarity, strength and consensus of opinions for each aspect. In making these decisions, the system

may use information about the user (e.g., the user's preferences regarding which entities and attributes to discuss), information from human-authored reviews (e.g., the words used in a collection of reviews), and information about the entities and aspects themselves (e.g., relationships between aspects of an entity).

The GEA system was one of the first abstractive summarizers: it generated user-tailored summary recommendations of houses (Carenini and Moore 2006). It used 14 aspects in the house buying domain, including exterior appearance, distances from various landmarks, and sizes of areas of the property such as the garden and porch. Given a model of a user's preferences across these 14 aspects, GEA would select a house from its database and recommend it to the user. The summary recommendation would focus on aspects that were *notably compelling*, i.e., with strong positive or negative values and important to the user. The summary recommendation would include justification statements for notably compelling aspects with positive values, and concession statements for notably compelling aspects with negative values.

The MATCH dialog system, an early multimodal mobile application for local business search, also included an abstractive summarizer (Johnston et al. 2001). Users of MATCH could search for restaurants in Manhattan that matched particular criteria (e.g. "cheap", "Italian", "on the Upper West Side"). Users could browse the retrieved restaurant set on a map, get directions to a selected restaurant, or read automatically generated, user-tailored summaries, comparisons or recommendations of restaurants (Walker et al. 2004). MATCH used seven aspects of restaurants: food quality, food type, cost, service, neighborhood, and décor. The rating values for each aspect were obtained from an independent restaurant rating service; there was only one value for each aspect for each restaurant, and there were no input text reviews. Given one or more restaurants to summarize, compare or recommend, MATCH would select restaurants and restaurant aspects based on a pre-specified model of the importance of each restaurant aspect to the target user: first, the model would be used to weight the aspect ratings for each aspect, and then the overall rating for the restaurant would be computed as a sum of the weighted aspect ratings. Restaurants with higher overall user-weighted quality would be explicitly mentioned in comparisons and summaries, and aspects with large (positive or negative) user-weighted ratings would be explicitly mentioned in recommendations, comparisons and summaries. A conciseness factor was used to keep the total content of any one presentation within bounds.

Evaluations of both MATCH and GEA showed that user-targeted presentations were preferred by users over generic or other-targeted presentations (Carenini and Moore 2006; Walker et al. 2004). In later work with the MATCH data, Higashinaka et al. looked at using a corpus of human-authored reviews to make decisions about how to construct and realize reviews (Higashinaka et al. 2006).

Carenini et al. have done several comparisons of extractive and abstractive summarizers, with mixed results. In a first study, the extractive summarizer was found to be more linguistically interesting, while the abstractive one was found to be more accurate; in terms of overall user's evaluation rating the two were equivalent (Carenini and Pauls 2006). In a more recent comparison specifically for reviews of

controversial items, the abstractive summarizer was rated as producing summaries with better linguistic and overall quality, but not more informative content (Carenini and Cheung 2008).

In Have2eat, a simple extractive summarizer is currently used. However, the weaknesses of extractive summarization (redundancy, less ability to co-constrain informativeness and conciseness, lack of coherence in generated summaries) have inspired us to consider a hybrid approach. We present this hybrid approach, which is a focus of our current research, in the next section.

A Modest Proposal for a Hybrid Solution to Review Synthesis

Extractive summaries are linguistically interesting and can be both informative and concise. They also require less engineering effort. On the other hand, abstractive summaries tend to have better coverage for a particular level of conciseness, and to be less redundant and more coherent. They also can be constructed to target particular discourse goals, such as summarization, comparison or recommendation. Although in theory, it is possible to produce user-targeted extractive summaries, user-specific review summarization has only been explored in the context of abstractive summarization.

There are additional features of review data that no review summarizer currently takes into account. These include temporal features (in particular, how old the reviews are – products and services may change over time) and social features (in particular, social or demographic similarities or relationships between reviewers and the reader of the summary). In addition, there is an essential contradiction at the heart of current review summarization systems: the system is authoring the review, but the opinions contained therein are really attributable to one or more human authors, and those attributions are not retained in the review summary. For example, in the STARLET review in Table. 11.10.

Do the two sentences about wings reflect one (repeated) opinion from a single reviewer, or two opinions from two separate reviewers? The ability to attribute subjective statements to known sources makes them more trustworthy; conversely, in the absence of the ability to attribute, a reader may become skeptical or confused about the content of the review summary. If we have as input to a review summary system a set of reviews, each sentence of which is labeled with aspect ratings (and associated confidence scores) and authorship information (at minimum, a unique identifier for each author), then we can create hybrid abstractive/extractive reviews that:

- Are informative – achieve broad coverage of the input opinions
- Are concise and avoid redundancy
- Are readable and coherent (of high linguistic quality)
- Can be targeted to the reader
- Solve the attribution problem

Table 11.10 Example of STARLET extractive summary

Delicious. Can't wait for my next trip to Buffalo.
GREAT WINGS.
I have rearranged business trips so that I could stop in and have a helping or two of their wings.
We were seated promptly and the staff was courteous.

The algorithm for a hybrid review summarizer is as follows:

1. If desired, weight per-sentence aspect ratings according to the timeliness of the review, the trustworthiness/reliability of the reviewer, the demographic similarity of the reviewer to the target reader, or other factors.
2. Compute (weighted) mean aspect ratings, aspect rating ranges, and aspect rating distribution curves, for each entity by summing the (weighted) per-sentence aspect ratings. A rating distribution curve may skew positive or negative, may be normal, may be uniform, or may be bimodal. For each aspect, construct a sentence "bin" and put all *quotable* sentences with (reliable) opinion ratings for that aspect in that aspect's bin.
3. Compute an overall (user-targeted) rating for each entity.
4. Select the entity, i.e., product or service, to include in the generated presentation.[11]
5. Decide what the summary claim will be for the selected entity. If the overall rating for the entity is strongly positive, the entity will be recommended. If it is strongly negative, the entity will be disrecommended. If it is neither negative nor positive, no summary claim will be generated.
6. Construct a preferred order of selection for the aspects based on the strength of the (weighted) overall aspect ratings for the selected entity. All other things being equal, strong positive and negative opinions will be selected before weaker opinions. If an aspect is controversial (has a bimodal distribution curve), its mean rating may not accurately reflect the range of reviewers' opinions, so controversial aspects should be preferred over noncontroversial weak opinion aspects. If there are comparatively few ratings for an aspect, the aspect may be less preferred.
7. For each aspect in the aspect preference list, select content to communicate about that aspect:
   - If the aspect skews positive, skews negative, or has a normal distribution curve, communicate the number (e.g., "most reviewers thought", "some reviewers thought"), mean and range of the aspect ratings. If there is not already a quote in the content selected so far that mentions this aspect, then select a quote from the aspect's bin that is strongly correlated with the mean rating for the aspect, and attach it as an optional elaboration to the aspect rating summary.

[11] For this discussion we assume each generated presentation is for only one entity; but comparisons, summaries and other types of presentation are also possible.

- If the aspect has a uniform distribution curve, communicate the number and range of the aspect ratings.
- If the aspect has a bimodal distribution curve, communicate the number or percent of ratings at each end of the curve. Select a quote from the aspect's bin that is strongly correlated with each end of rating range for the aspect, and attach these as optional elaborations to the aspect rating summary.

Aspects with positive ratings *support* recommendations and are *concessions* for disrecommendations. The opposite is true for aspects with negative ratings. The selected content for each aspect should be labeled with that aspect's preference order.

At the end of this process, we have: (1) selected content elements for the presentation to be generated; (2) discourse relations between the selected content elements; (3) preference order information that can be used to prune the presentation to a desired level of conciseness. In the generated presentation, the order of presentation of aspects may be chosen by the preference order, or by using the most frequent order from the input reviews. Quotes can be set off by attribution statements such as "One reviewer commented" or "As Sally123 said" (and linked to the corresponding input reviews). Decisions about whether to include selected quotes in the final presentation, and about whether to include less-preferred attributes, can be made to optimize for conciseness or informativeness. Not all sentences are quotable. For example, some reviewers tell stories about their experiences and the sentences in these stories do not stand alone. A *quotable* sentence: is short (a simple independent clause), mentions only one or two aspects, and is interesting (compare "The food was good", which will just restate the aspect rating summary statement, with "I particularly liked the chicken wings", which contains an example). In our preliminary experiments, we are using supervised learning techniques to train a binary classifier using ngram, part of speech and chunk features to identify quotable sentences. In second step, using a similar feature set, we classify aspects and opinion polarity associated to each quote.

This hybrid approach builds on the innovations of previous abstractive and extractive summarizers, combining the best of both approaches to produce informative, concise, accurate and readable summaries. We are currently implementing and testing this approach and plan an evaluation in the restaurant review domain.

Conclusions

Recent years have seen a dramatic increase in the use of networked mobile devices for daily life tasks such as finding and purchasing products and services. This kind of task involves search (to retrieve matching entities or businesses) and selection (of retrieved results). Selection decisions are increasingly made on the basis of surveying online reviews. However, mobile users lack time and display space for leisurely

perusal of hundreds (or thousands) of reviews for dozens of search results. Sentiment analysis and review summarization are therefore enabling technologies for these tasks – they can increase user satisfaction and productivity. In this chapter, we have reviewed the state of the art in sentiment analysis and review summarization. We compared the best methods for multi-aspect sentiment analysis. We surveyed graphical, extractive and abstractive approaches to review summarization, and sketched a promising hybrid approach. We presented a mobile application, Have2eat, that contains implementations of sentiment analysis and review summarization, and that illustrates the promise of these technologies.

Acknowledgments We thank Barbara Hollister, Narendra Gupta, Jay Lieske, Sveva Besana, and Kirk Boydston, for their contributions and great enthusiasm.

References

Baccianella S, Esuli A, Sebastiani F (2009) Multi-facet rating of product reviews. In: Proceedings of the 31th european conference on IR research on advances in information retrieval, ECIR '09. Springer, Berlin/Heidelberg, pp 461–472

Berger AL, Pietra VJD, Pietra SAD (1996) A maximum entropy approach to natural language processing. Comput Linguist 22:39–71

Carenini G, Cheung JCK (2008) Extractive vs. NLG-based abstractive summarization of evaluative text: the effect of corpus controversiality. In: INLG '08: proceedings of the fifth international natural language generation conference. Association for Computational Linguistics, Stroudsburg, pp 33–41

Carenini G, Moore JD (2006) Generating and evaluating evaluative arguments. Artif Intell 170(11):925–952

Carenini G, Ng R, Pauls A (2006) Multi-document summarization of evaluative text. In: 11th meeting of the European chapter of the association for computational linguistics (EACL 2006). The Association for Compute Linguistics, Stroudsburg

Carenini G, Cheung J, Pauls A (2012) Multi-document summarization of evaluative text. Comput Intell (to appear)

Chevalier JA, Mayzlin D (2006) The effect of word of mouth on sales: online book reviews. J Mark Res 43(3):345–354

Conroy JM, Stewart JG, Schlesinger JD (2005) CLASSY query-based multi-document summarization. Proceedings of the document understanding conference Wksp. 2005 (DUC 2005) at the human language technology conf./conf. on empirical methods in natural language processing (HLT/EMNLP). The Association for Computational Linguistics, Stroudsburg

Copeck T, Japkowicz N, Szpakowicz S (2002) Text summarization as controlled search. In: Proceedings of the 15th conference of the Canadian society for computational studies of intelligence on advances in artificial intelligence, AI '02. Springer, London, pp 268–280

Crammer K, Singer Y (2001) Pranking with ranking. In: Dietterich TG, Becker S, Ghahramani Z (eds) Neural information processing systems (NIPS), Vancouver, British Columbia. MIT, Cambridge, pp 641–647

Dave K, Lawrence S, Pennock DM (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW '03: proceedings of the 12th international conference on World Wide Web. ACM, New York, pp 519–528

Di Fabbrizio G, Aker A, Gaizauskas R (2011) STARLET: multi-document summarization of service and product reviews with balanced rating distributions. In: Proceedings of the 2011 IEEE

international conference on data mining (ICDM) workshop on sentiment elicitation from natural text for information retrieval and extraction (SENTIRE), Vancouver, Canada
Duan W, Gu B, Whinston AB (2008) Do online reviews matter? – An empirical investigation of panel data. J Decis Support Syst 45(4):1007–1016
Elhadad N, Kan MY, Klavans JL, McKeown KR (2005) Customization in a unified framework for summarizing medical literature. Artif Intell Med 33:179–198
Feng S, Xing L, Gogar A, Choi Y (2012) Distributional footprints of deceptive product reviews. In: Breslin JG, Ellison NB, Shanahan JG, Tufekci Z (eds) International AAAI conference on weblogs and social media. AAAI, Menlo Park
Goldstein J, Mittal V, Carbonell J, Kantrowitz M (2000) Multi-document summarization by sentence extraction. In: Proceedings of the 2000 NAACL-ANLPWorkshop on automatic summarization – volume 4, NAACL-ANLP-AutoSum '00. Association for Computational Linguistics, Stroudsburg, pp 40–48
Haffner P, Tur G, Wright JH (2003) Optimizing SVMs for complex call classification. In: 2003 IEEE international conference on acoustics, speech, and signal processing, 2003. Proceedings (ICASSP '03). Institute of Electrical and Electronics Engineers Inc. vol 1, pp I–632–I–635
Haffner P, Phillips SJ, Schapire RE (2005) Efficient multiclass implementations of l1-regularized maximum entropy. CoRR abs/cs/0506101
Higashinaka R, Prasad R, Walker MA (2006) Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In: Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics, ACL-44. Association for Computational Linguistics, Stroudsburg, pp 265–272. doi:http://dx.doi.org/10.3115/1220175.1220209
Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD). DBLP, pp 168–177
Johnston M, Bangalore S, Vasireddy G, Stent A, Ehlen P, Walker M, Whittaker S, Maloor P (2001) Match: an architecture for multimodal dialogue systems. In: ACL '02: proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, Morristown, pp 376–383
Ku LW, Liang YT, Chen HH (2006) Opinion extraction, summarization and tracking in news and blog corpora. In: Proceedings of AAAI-2006 spring symposium on computational approaches to analyzing weblogs
Li L, Lin HT (2007) Ordinal regression by extended binary classification. In: Schölkopf B, Platt JC, Hofmann T (eds) Advances in neural information processing systems 19. MIT, Cambridge, pp 865–872
Lin D (1998) Automatic retrieval and clustering of similar words. In: COLING-ACL, pp 768–774
Lin CY (2004) ROUGE: a package for automatic evaluation of summaries. In: Proceedings of ACL workshop on text summarization branches out, p 10
Liu J, Seneff S, Zue V (2010) Utilizing review summarization in a spoken recommendation system. In: Proceedings of SIGDIAL
Lu Y, Zhai C, Sundaresan N (2009) Rated aspect summarization of short comments. In: WWW '09: proceedings of the 18th international conference on World Wide Web. ACM, New York, pp 131–140. doi:http://doi.acm.org/10.1145/1526709.1526728
Lu B, Ott M, Cardie C, Tsou BK (2011) Multi-aspect sentiment analysis with topic models. In: Spiliopoulou M, Wang H, Cook DJ, Pei J, Wang W, Zaïane OR, Wu X (eds) 2011 IEEE 11th international conference on data mining workshops (ICDMW), Vancouver, BC, Canada, 11 Dec 2011. IEEE
McDonald R, Hannan K, Neylon T, Wells M, Reynar J (2007) Structured models for fine-to-coarse sentiment analysis. In: Proceedings of the association for computational linguistics (ACL). Association for Computational Linguistics, Prague, pp 432–439
McKeown KR, Barzilay R, Evans D, Hatzivassiloglou V, Klavans JL, Nenkova A, Sable C, Schiffman B, Sigelman S (2002) Tracking and summarizing news on a daily basis with

columbia's newsblaster. In: Proceedings of the second international conference on human language technology research, HLT '02. Morgan Kaufmann, San Francisco, pp 280–285
Mithun S, Kosseim L (2009) Summarizing blog entries versus news texts. In: Proceedings of the workshop on events in emerging text types, eETTs '09. Association for Computational Linguistics, Stroudsburg, pp 1–8
Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the ACL, pp 271–278
Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135
Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), pp 79–86
Park D, Lee J, Han I (2007) The effect of on-line consumer reviews on consumer purchasing intention: the moderating role of involvement. Int J Electron Commer. 11:125–148. doi:10.2753/JEC1086-4415110405
Polanyi L, Zaenen A (2005) Contextual valence shifters. In: Computing attitude and affect in text. Springer, Dordrecht
Radev D, Allison T, Blair-Goldensohn S, Blitzer J, Çelebi A, Dimitrov S, Drabek E, Hakim A, Lam W, Liu D, Otterbacher J, Qi H, Saggion H, Teufel S, Topper M, Winkel A, Zhang Z (2004) MEAD – a platform for multidocument multilingual text summarization. In: Conference on language resources and evaluation (LREC), Lisbon, Portugal
Saggion H, Lapal G (2002) Generating indicative-informative summaries with sumum. Comput Linguist 28:497–526 doi:http://dx.doi.org/10.1162/089120102762671963
Sarle WS (1994) Neural networks and statistical models. In: Proceedings of the nineteenth annual SAS users group international conference, Apr 1994. SAS Institute, Cary, pp 1538–1550
Schapire RE, Singer Y (2000) Boostexter: a boosting-based system for text categorization. Mach Learn 39(2/3):135–168
Shimada K, Endo T (2008) Seeing several stars: a rating inference task for a document containing several evaluation criteria. In: Advances in knowledge discovery and data mining, 12th Pacific-Asia conference, PAKDD 2008. Lecture notes in computer science, Osaka, vol. 5012. Springer, Berlin/New York, pp 1006–1014
Snyder B, Barzilay R (2007) Multiple aspect ranking using the good grief algorithm. In: Human language technologies 2007: the conference of the North American chapter of the association for computational linguistics; proceedings of the main conference, Rochester, New York. Association for Computational Linguistics, Stroudsburg, pp 300–307
Stent A, Zeljković I, Caseiro D, Wilpon J (2009) Geo-centric language models for local business voice search. In: Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the association for computational linguistics, NAACL '09. Association for Computational Linguistics, Stroudsburg, pp 389–396
Stone PJ, Dunphy DC, Smith MS, Ogilvie DM (1966) The general inquirer: a computer approach to content analysis. MIT, Cambridge
Titov I, McDonald R (2008) Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th international conference on World Wide Web, WWW '08. ACM, New York, pp 111–120
Walker MA, Whittaker SJ, Stent A, Maloor P, Moore JD, Johnston M, Vasireddy G (2004) Generation and evaluation of user tailored responses in multimodal dialogue. Cogn Sci 28(5):811–840
Wiebe J, Wilson T, Bruce R, Bell M, Martin M (2004) Learning subjective language. Comput Linguis 30(3):277–308

Chapter 12
Mobile Speech and the Armed Services: Making a Case for Adding Siri-like Features to VAMTA (Voice-Activated Medical Tracking Application)

James A. Rodger and James A. George

Abstract In this chapter we take a look at how to improve VAMTA (voice-activated medical tracking application), a program we introduced several years ago which has been successfully adopted by the military, by adding natural language capabilities that would enable VAMTA to perform as a personal assistant and knowledge navigator in the medical-military mobile environment. We consider some of the key functions of a Siri-enhanced VAMTA, which would use a natural language interface to answer questions, make recommendations and perform actions by delegating requests to a set of Web services. We explore the use of fuzzy linguistic ontologies for natural language applications, which would enable this natural language driven medical tracking program to fulfill a wide range of tasks for military personnel in a mobile setting.

Introduction

The Voice Activated Military Tracking Application (VAMTA) has been studied in two phases: The first was an initial feasibility study of the practicalities of using a speech-driven tracking application. The second was a study of how end-users perceived VAMTA in terms of how likely they were to accept this technology as

J.A. Rodger (✉)
Indiana University of Pennsylvania, MIS and Decision Sciences, Eberly College of Business and Information Technology, 664 Pratt Drive, Indiana, PA 15705, USA
e-mail: jrodger@iup.edu

J.A. George
Business, Economy and Politics Columnist, Examiner.com, 4650 Washington Blvd 1008, Arlington, VA 22201, USA
e-mail: jagpr.net@gmail.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_12,
© Springer Science+Business Media New York 2013

part of their daily routine (Rodger and George 2010).[1] In this chapter we take a look at how to improve VAMTA by adding natural language capabilities that would enable VAMTA to perform as a personal assistant and knowledge navigator in the medical-military mobile environment. We consider some of key functions of a Siri-enhanced[2] VAMTA, which we refer to as "SIRVAMTA" (the "SIR" stands for Siri-integrated response). This mobile tracking application would use a natural language interface to answer questions, make recommendations and perform actions by delegating requests to a set of Web services. We explore the use of fuzzy linguistic ontologies for natural language applications, which would enable SIRVAMTA to fulfill a wide range of tasks for military personnel in the mobile environment.[3]

The way this would work is that the enterprise intelligence web (EIW) would allow the speech recognition natural language feature, or Siri, that is be built into VAMTA to access web services in the DoD (Department of Defense) cloud. In so doing, the enterprise architecture and fuzzy linguistic ontologies would document, store and analyze the medical event to an alerting subsystem while information extraction and automated semantic reasoning would be used to access open source data for medical news feeds, public web pages and RSS. In addition, mission data and sensor data such as xml csv can piggyback on the mobile medical data so that the resulting semantics would relate unstructured data to structured data in the correct structure that effectively reduces information overload to the medical team.

Since VAMTA was designed to address information overload by enhancing the electronic management of patient data, incorporating the latest NL recognition fuzzy ontologies would constitute the next big step for VAMTA. In this chapter we will provide a specific approach to fuzzy linguistic ontologies that demonstrates how Siri can be applied to VAMTA in the mobile medical-military environment. But first we examine briefly why ontologies are important in the first place.

---

[1] The original feasibility literature, circa 2000, evolved from the initial reporting of the VAMTA findings (Rodger and Pendharkar 2007) to reporting of the end-user perceptions of the VAMTA task-technology fit and the smart-data strategy for optimization of performance Rodger, J. A. and George, J. (2010). Note; citations are placed in references and thus are never placed in footnote other than the authors name and date.

[2] We use Siri in the generic sense to refer the use of a personal assistant that understands natural language commands. In truth we could use Nina, designed by Nuance, or any other kind of personal assistant model for that matter. However, for the purposes of this discussion we use the term "Siri" which became known to the public as the first natural language driven mobile device when Apple unveiled the Siri feature on its 4S iPhone last fall.

[3] While we are cognizant of the fact that natural language shortcomings still plague voice recognition applications (Scharenborg 2007; Siniscalchi and Lee 2009; Cooke et al. 2009), we are nevertheless inspired to utilize natural language in the VAMTA.

Why Ontologies Are Important in Designing and Implementing Natural Language Software for Both Commercial and Military Applications

To begin with, natural language is particularly important in the medical-military environment because it can improve the healthcare delivery process by helping information seekers to retrieve relevant, accurate and timely data that inform decision makers and help drive medical actions. Below, we briefly summarize a few studies on the advances in both design and implementation of natural language (NL) software to support our proposition about the added value of natural language in the medical-military mobile environment. For example, Ionita (2008) presents a method of implementing the voice recognition for the control of software applications. The solutions proposed are based on transforming a subset of the speech phonemes into NL commands that are recognized by the application using a formal language defined by means of a context free grammar. The work of this author also included results on the integration of modality in voice recognition with voice synthesis for the Romanian language in a Windows application.

Schorlemmer and Kalfoglou (2008) address the importance of ontologies pointing to the scarcity of general mathematical foundations for ontology-based semantic integration underlying current knowledge engineering methodologies in decentralized and distributed environments. After reviewing the first-order ontology-based approach to semantic integration and a formalization of ontological commitment, they proposed a general theory that uses a syntax- and interpretation-independent formulation of language, ontology and ontological commitment in terms of institutions. Their thesis is that their formalization generalizes the intuitive notion of ontology-based semantic integration while retaining its basic insight, which they use to elicit and hence compare various increasingly complex notions of semantic integration and ontological commitment based on differing understandings of semantics.

Sleeman et al. (2008) developed the Designers' Workbench for large companies, such as Rolls-Royce, to ensure that the design of each product is consistent with the specification for the particular design as well as with the company's design rule books. The evolving design of a product is described against the jet engine ontology. Design rules are expressed as constraints over the domain ontology. To capture the constraint information, a domain expert such as a design engineer, has to work with a knowledge engineer to identify the constraints, and it is then the task of the knowledge engineer to encode these into the Workbench's knowledge base. This is undoubtedly an error-prone and time consuming task. To relieve the knowledge engineer of the tedium of this task the authors have developed a tool (ConEditor+) that enables domain experts themselves to capture and maintain these constraints. The tool allows the user to combine selected entities from the domain ontology with keywords and operators of a constraint language to form a constraint expression. In order to appropriately apply, maintain and reuse constraints, the authors believe that it is important to understand the assumptions and context in which each constraint is applicable. They refer to these as "application conditions" and hypothesize that an explicit representation of constraints

together with the corresponding application conditions and the appropriate domain ontology could be used by a system to support the maintenance of constraints. In this paper, the authors focus on the important role that the domain ontology plays in supporting the maintenance of constraints in engineering design.

Massie et al. (2008) point out that voice automated computing and speech recognition technology are beginning to revolutionize the commercial industry as speech recognition systems are becoming widely used in many real world applications, such as commercial banking and airline reservations. The authors show the utility of speech recognition technology to support the command, control and information-fusion needs of dismounted soldiers engaged in specialized tactical operations. Their application is presented in terms of an operational and systems architecture, which includes a vocabulary of grammar and sample voice choreography. These artifacts are used to illustrate autonomous voice access, which is defined as a soldier's ability to voice authenticate, access, search and retrieve tactical information assets from backend systems equipped with speech recognition capabilities. The authors of this study believe that as voice and data networks continue to converge, speech recognition and interactive voice response (IVR) technology will drive the evolution of voice-enabled tactical communication portals. This would enable enabling soldiers to remotely access information through specialized voice enterprise services.

Launching SIRVAMTA in the Medical-Military Mobile Setting

We are targeting the Joint Military Medical Command of the US Department of Defense as our prospective customer with the goal of validating that VAMTA can be made more effective in achieving certain tasks, such as answering questions, making recommendations and delegating requests to a set of Web services, by incorporating cutting-edge natural language applications such as Siri. By adding personal-assistant features, VAMTA will then be able to bring up the weather report at an accident scene. In addition, it will also allow access to GPS navigation so that health care workers and paramedics can arrive at the accident scene, using the quickest and most effective route possible. Our goal would be to disseminate VAMTA throughout the medical-military community on a mobile platform by adopting a fuzzy linguistic ontology algorithm (FLOMA), which helps to standardize data capture, recording and processing.

Once these fuzzy linguistic ontologies are entered into an editor such as Protégé, the system designers will be able to use the SIRVAMTA to perform multiple mobile tasks in the medical- military setting. For example, you can search the web for protein examples by saying "search protein" (Gadchick 2011). You may also specifically call a browser for your search by saying "Google protein". You can say the word "note" and make a note of the protein search. In additions, Siri can then be used to check, send and reply to your e-mails on proteins in the medical diagnosis thread. This of course adds to the routine function of Siri-like features which would provide VAMTA with the user's present location, corrects his grammar and allows the user to text hands-free.

Here is an example of how the Siri MedCon Physian-Patient Communications App might work:

Doctor: "Tell [Nurse Jones] I'll be right there"
SIRVAMTA : "Ok. Calling Nurse Jones"
Doctor: "Send a message to [Nurse Jones]"
SIRVAMTA : "Ok. Sending a message to Nurse Jones"
Doctor: "Text [Nurse Jones] [Administer 10 cc of Ringer's lactate STAT]"
SIRVAMTA : "Ok. Texting administer 10 cc of Ringer's lactate STAT"
Doctor: "Access Jane Smith's record for the past 10 days."
SIRVAMTA : "Ok. Here is Jane's record and five questions to ask during her visit."

While SIRVAMTA is still in its conception stage, the example above shows the kinds of possibilities of this mobile platform application. At the recent Association for Enterprise Integration (AFEI) conference[4] we presented our POC (proof of concept) for SIRVAMTA, supporting our argument for pilot testing SIRVAMTA by pointing to the progress made in medical tracking applications which had once been performed entirely on a laptop computer. We showed in contrast that in today's world with medical tracking being performed on compact mobile devices the environment may be ripe for launching SIRVAMTA in the mobile medical- military setting. Without doubt there is a very practical reason for rapid adoption of SIRVAMTA: the faster that caregivers adopt this technology, the faster that standardization can occur; this can only result in improved health outcomes and the saving of lives.

DoD Business Enterprise Architecture and Web-Based Ontologies

The DoD Business Enterprise Architecture (BEA) has dedicated itself to business-process redesign and the building of an architecture that will enable cost accountability and drive out redundancy (Wisnosky 2012). This DoD BEA project called for a progression from a blueprinted readable, but not analyzable, stovepipe mentality to a semantic Smart-Data end-to- end based approach that is executable. The Smart Data approach adopted the concepts of enterprise performance optimization through interoperability and the application of correct methods and algorithms. In year 2010 the DoD began to use the Smart-Data approach to medical tracking in VAMTA.[5] The road ahead requires data integration, business intelligence, a common vocabulary

[4] Association For Enterprise Integration (AFEI) conference, in Miami, on April 30-May 3, 2012.

[5] "Smart Data" is a unique concept formulated by the author and his colleague, Jim George George and Rodger, (2010). The goal of Smart-Data is to promote a new approach to consulting and business operations. Smart Data has three dimensions: (1) enterprise performance; (2) the application of metrics and algorithms; and (3) interoperability within the organization. All three dimensions are consequently interwoven. To wit, enterprise performance is achieved through the application of metrics and algorithms which promotes interoperability within the organization.

ontology, rules and security. This process also involves enabling the architecture, integrating the data, building an enterprise information web and embedding DoD policy in a net-centric approach that utilizes ontologies on the Semantic Web.[6]

Fuller (2008) points out that ontology consists of a hierarchical description of important classes or concepts in a particular domain, along with the description of the properties of the instances of each concept. Web content is then annotated by relying on the concepts defined in specific domain ontology. An ontology is a specification of conceptualization and a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge level agent is committed to some conceptualization, explicitly or implicitly.

Ontologies are important because the Web Ontology Language (OWL) is an industry standard that is backed by the WC3 open specification. OWL is a family of knowledge representation languages for authoring ontologies and it is endorsed by the World Wide Web Consortium. This family of languages is based on two semantics: OWL DL and OWL Lite semantics; both are based on Description Logics, which is a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. Today DL has become a cornerstone of the Semantic Web for its use in the design of ontologies.

Editors and tools such as Protégé, as mentioned in the prior section, exist that can create and visualize OWL files and process OWL rules. This is supported by an active growing community that supports these advances and utilizes formal logic and reasoning to support capturing rules and data in a controlled fashion. Ontologies also support the Smart Data concept of interoperability: First, because OWL files are easily shared making structures visible and well understood. Second, queries of specific data solutions can be provided by SPARQL which is a recursive acronym for the SPARQL Protocol and Resource Description Framework (RDF) Query Language (Wikipedia SPARQL 2012). Third, same concepts with different terms across federated ontologies can be resolved via OWL "sameAs" constructs.

VAMTA fits nicely into this process by making authoritative data accessible and understandable for humans to analyze and machines to manipulate so that the DoD can deliver medical treatment in a mobile setting. Through this architecture, VAMTA can be redesigned to utilize NL processors such as Siri to discover and consume data by providing data analytics which are used for problem-solving. Another function of SIRVAMTA is to get data to the medics in the field in a form that is usable in the field environment we also need to add NL sensory awareness properties such as "who are you?" "where are you?" "what do you want?" to this updated personal military, mobile, medical assistant.

[6] The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web content. The Web is considered as a universal medium for data, information and knowledge exchange.

The Fuzzy Linguistic Ontology Domain Model and Resources, Events and Agents

Ontologies are of particular importance in the fields of computer science and information systems because of their ability to model/represent knowledge both as a set of concepts and the relationships between these concepts, in a given domain. To wit, if ontology is formulated in a crowd-sourced/collaborative manner using a formal language then one can arrive at a formal unambiguous model/representation of the knowledge, referred to as ontology, about a given domain. Ontologies have developed for a variety of tasks like enterprise modeling (Gruninger et al. 2000), classifying in-vivo biological cell types (Meehan et al. 2011), marketing in relation to brand management (Grassl 1999), and socialism (Westra 2002). It is well understood that that a modern enterprise must be data driven and all decisions should be based on information (George and Rodger 2010). The process of preparing data for both transformation of the data into information and presenting the data for action, depends upon ontology. Furthermore, the process of associating information with experience, methods, and algorithms likewise depends on ontology (Zadeh and Kacprzyk, 1992). Thus, when given a problem domain within the context of an enterprise the first step would be to represent the domain using an ontology. The ontology representation of syntax and semantics used to state the concepts and their relationships should be based on standards, especially if the domain spans across several enterprises.

Trappey et al. (2009) present a novel hierarchical clustering approach for knowledge document self-organization, particularly for patent analysis. Carlsson and Fuller (2011) point out that fuzzy ontologies have been proposed as a solution for addressing semantic meaning in an uncertain world, where reasoning is approximate rather than precise. Fuzzy ontologies avoid many of the weaknesses of their original counterparts by facilitating interoperability between independent ontologies and their databases (Cross 2004). In short, an ontology is a conceptualization of a domain into an understandable human machine-readable format. This format consists of entities, attributes, relationships and axioms (Guarino and Giarretta 1995).

REA (Resources, Events and Agents) is described in Wikipedia[7] as the model that is used in an Accounting Information System (AIS). This is important to NL processing, because much like the REA model can effectively eliminate many accounting objects that are not necessary, such as debits and credits, the computer can likewise generate NL in real time using only the pertinent objects, while eliminating what is not relevant. REA treats the system as a virtual representation of the actual medical process by creating computer objects that directly represent real-world-medical objects. In essence, REA is an ontology and the real objects included in the REA medical model are the resources of medical services, the events of patient encounters and the agents of healthcare professionals. Within the REA model is also a pair of events that are linked by an exchange relationship otherwise known as a "duality" relation. This relationship represents a resource that is given away or lost, such as a healthcare worker's time, and a resource that is received or gained, such as a physical diagnosis. Of course this relationship in

[7] http://en.wikipedia.org/wiki/Resources,_events,_agents_(accounting_model).

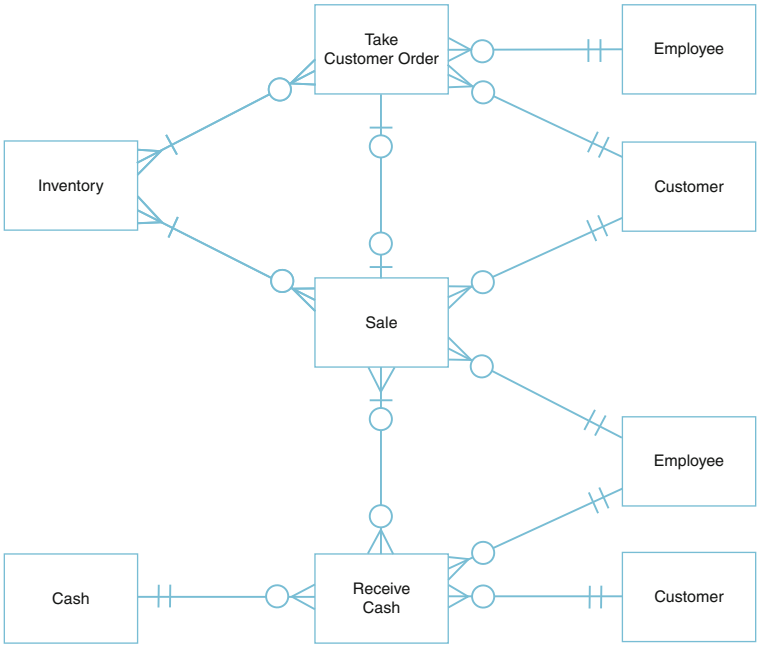

Fig. 12.1 Example of the AIS basic REA pattern (Wikipedia 2012)

reality is more complex and often involves more than two events. REA ontology systems are usually modeled as relational databases, and the software design typically uses entity-relationship diagrams. This philosophy of REA draws on the software engineering concept of reusable Design Patterns. REA patterns are used to describe databases rather than object oriented programs such as those employed in Unified Modeling Language (UML). Figure 12.1, below, presents an illustrative AIS example. The AIS model will evolve into a medical example later in the paper.

Demonstration of Fuzzy Linguistic Ontology

Fuller (2008) points out that OWL DL becomes less suitable in domains in which the concepts that are represented do not avail themselves of short, precise and narrow definitions. When dealing with Web content, however, this situation in which concepts do not fit into narrow definitions is likely to be the rule rather than an exception. For instance, just consider our case in which we would like to build an ontology about proteins using the Siri application in our VAMTA device. We may encounter the problem of representing concepts like “Proteins are polypeptide amphipathic molecules composed of hydrogen and carbon atom polymers”. It soon becomes apparent that such concepts can hardly be encoded into OWL, as they involve fuzzy or vague concepts, like “hydrogen atoms”, “carbon atoms”, “polymers”, “polypeptide” and

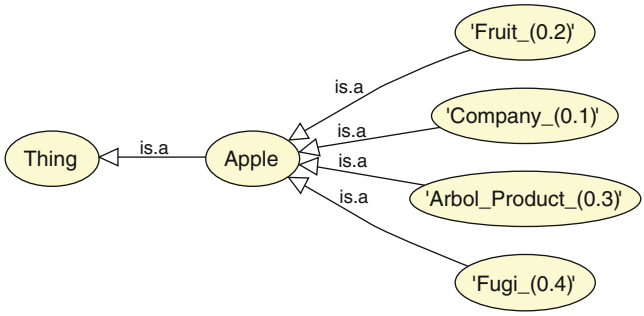

Fig. 12.2 Probability of the linguistic occurrence for the thing "apple" (Parry 2005)

"amphipathic", for which a clear and precise definition is impossible. We address this problem by assigning weights to a fuzzy ontology to determine the probability of the linguistic occurrence [As shown in Fig. 12.2, below, of the "apple" example] (Parry 2005). A fuzzy ontology is a quintuple F=< I, C,T,N,X>where:

- I is the set of individuals (objects), also called instances of the concepts.
- C is a set of fuzzy concepts (or classes—cf. in OWL—of individuals, or categories, or types).
- Each concept is a fuzzy set on the domain of instances.
- The set of entities of the fuzzy ontology is defined by E=C I.
- T denotes the fuzzy taxonomy relations among the set of concepts C.
- It organizes concepts into sub-(super-) concept tree structures.
- The taxonomic relationship T (i, j) indicates that the child j is a conceptual specification of the parent i with a certain degree.
- N denotes the set of non-taxonomy fuzzy associative relationships that relate entities across tree structures, for example:
  - Naming relationships, describing the names of concepts
  - Locating relationships, describing the relative location of concepts
  - Functional relationships, describing the functions (or properties) of concepts
- X is the set of axioms expressed in a proper logical language, i.e., predicates that constrain the meaning of concepts, individuals, relationships and functions.
- If a unit is included in the sample with probability Pi, then its base weight, denoted by wi, is given by wi=1/ pi.

This example could then be adapted to medical terminology and linguistics.

Protégé Fuzzy Linguistic Ontology Object Parent and Child Example

The discussion above leads to the following definitions for the fuzzy linguistic variable ontology (Li et al. 2005): (Basic fuzzy ontology)—A basic fuzzy ontology is a 4-tuple $O_{F=}(c_a, C_F, F, U)$, where c_a, C_F, F, U, which satisfy the following conditions:

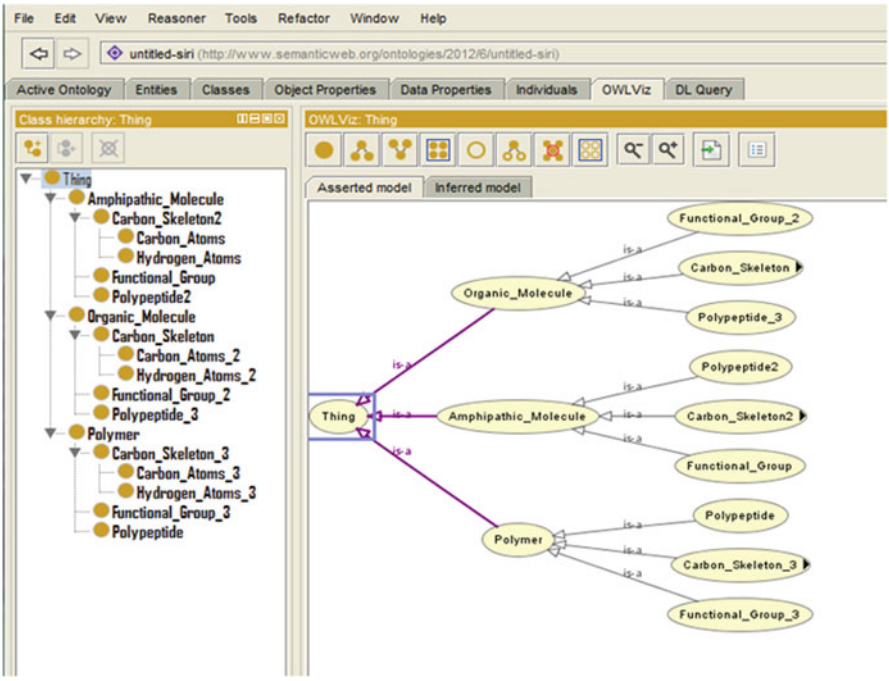

Fig. 12.3 Protégé fuzzy linguistic ontology object parent protein "thing"

1. $C_{F=}\{c_1,c_2,\ldots,c_n\}$ is a limited set.
2. Only one relation of this set, the relation of disjointedness, exists in C_F and C_F is complete at U. In the other words, C_F is a fuzzy partition of U.
3. C_F has an ordered relation $\leq$, and $<C_F,\leq>$ is a complete ordered set, i.e. all concepts in C_F constitutes a chain $c_1 \leq c_2 \leq \ldots \leq c_n$.
4. F is optional element of ontology.

An example of basic fuzzy ontology is $O_F = (\ c_a = \text{thing}, = C_F\ \{$ hydrogen atom polymer, carbon atom organic molecule, polypeptide amphipathic molecule$\}$, $U = [0,100]$. The combination of these fuzzy sets lead to some very specific linguistics.

The Protégé Fuzzy Linguistic Ontology object parent protein "thing" that was discussed above, is graphically illustrated in Fig. 12.3 and the child of carbon skeleton, which is "carbon atom" and "hydrogen atom" is demonstrated in Fig. 12.4.

Medical Subject Headings (MeSH) is a taxonomic hierarchy of medical and biological terms suggested by the U.S National Library of Medicine (NLM). Each term in MeSH has a unique tree number and several relating entry terms (Lowe and Barnett 1994). A tree number indicates the location of the term, and an entry term is either a synonym or an antonym for the corresponding term. AIMed is

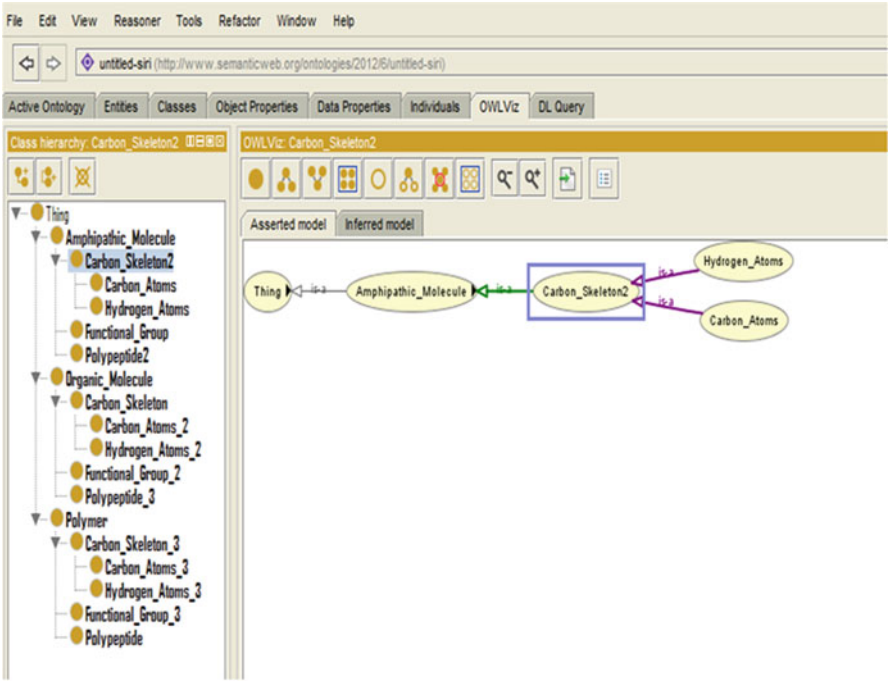

Fig. 12.4 "Carbon atom" and "hydrogen atom" child of "carbon skeleton"

derived from the Medical Literature Analysis and Retrieval System (MEDLINE) abstracts, and MEDLINE is manually indexed with MeSH vocabulary (Hliaoutakis 2005). MeSH and Artificial Intelligence Medicine (AIMed) are closely related (Bunescu et al. 2005). WordNet, is a well-known upper ontology that stores rich useful semantic information (Fellbaum 2010). Entries in WordNet are more general than those in MeSH. We can assume that the same contexts indicates the same relationships and so we can take advantage of WordNet to measure the semantic similarity of contexts such as the "proteins" example, because contexts often contain domain-unrelated words indexed by WordNet. Often semantic similarity measures can be partitioned into either how close the two concepts in the taxonomy are or on how much information the two concepts share. Figure 12.5 shows an example of MeSH taxonomy where both neurologic manifestations and pain are the ancestor of headache and neuralgia, and pain is the lowest common ancestor (LCA) of headache and neuralgia.

As mentioned previously, the REA model utilizes resources, events and agents to model relationships between objects. An ontology (O) organizes domain knowledge in terms of concepts (C), properties (P), relations (R) and axioms (A), and is formally illustrated in Fig. 12.6 as a mobile medical-military example:

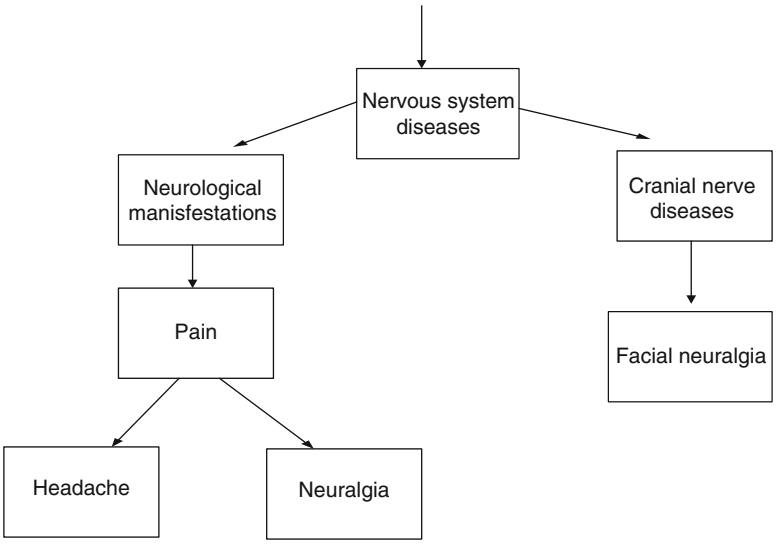

Fig. 12.5 A fragment of the MeSH taxonomy for nervous system diseases

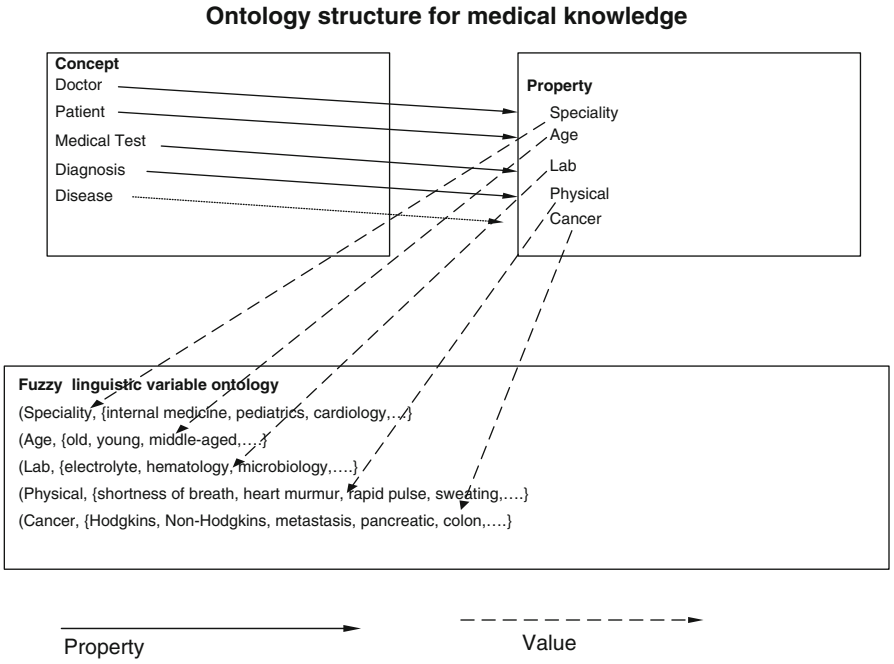

Fig. 12.6 Ontology structure for medical knowledge

Discussion

There is no paucity of data in the healthcare system. We look toward SIRVAMTA as a ripe opportunity for health practitioners to put large amounts of existing data to good use. In addition, SIRVAMTA can be instrumental in the collection of relevant new data that would make healthcare delivery more efficient in the mobile military setting. All in all, adding a personal assistant to VAMTA would serve as the basis for the creation of a medical-specific data engine for identifying the best data sources from the healthcare industry. In practical terms what one can envision is that the Siri application tied to VAMTA would offer a powerful combination of technology compacted into a mobile military healthcare package that can help patients and caregivers locate doctors, make appointments, and search for peer-to-peer community support. Furthermore, just as much as SIRVAMTA would serve the needs of lay members of the military, this personal assistant would likewise serve the needs of medical professionals who need to consult with specialists, give time-sensitive instructions to nurses, check for drug interactions, manage mortality and morbidity rates, and access electronic medical records in real time on a viable mobile platform. As we can see from these examples given here, a Siri powered VAMTA would allow at the very minimum access to relevant, time-sensitive data, while building a patient profile in real-time. These improvements to VAMTA would no doubt bring healthcare professionals one step closer to providing optimal care for the patients they serve.

References

Bunescu R, Ge R, Kate RJ, Marcotte EM, Mooney RJ, Ramani AK, Wong YW (2005) Comparative experiments on learning information extractors for proteins and their interactions. Artif Intell Med (Summarization and Information Extraction from Medical Documents) 33:139–155

Carlsson C, Fuller R (2011) Possibility for decision: a possibilistic approach to real life decisions. Springer, Berlin/Heidelberg

Cooke M et al (2009) Monaural speech separation and recognition challenge. Comput Speech Lang, (in Press). Corrected Proof, Available online 27 Mar 2009

Cross V (2004) Fuzzy semantic distance measures between ontological concepts. In: Proceedings of IEEE annual meeting of the North American fuzzy information processing society (NAFIPS 2004) Banff, June 27–30

Fellbaum C (2010) Wordnet, theory and applications of ontology: computer applications. Berlin, Springer, pp 231–243

Fullér R (2008) What is fuzzy logic and fuzzy ontology? KnowMobile national workshop, Helsinki, 30 Oct 2008

Gadchick (2011) The unofficial Siri handbook: the essential reference for your iPhone 4S. Amazon, New York

George JA, Rodger JA (2010) Smart data: enterprise performance optimization strategy. Wiley, New Jersey

Grassl W (1999) The reality of brands: towards an ontology of marketing. Am J Econ Sociol 58(2):313–359

Gruninger M, Atefi K et al (2000) Ontologies to support process integration in enterprise engineering. Comput Math Organ Theory 6:381–394

Guarino N, Giarretta P (1995) Ontologies and knowledge bases: towards a terminological clarification. In: Toward very large knowledge bases: knowledge building and knowledge sharing. Ios Press, Amsterdam

Hliaoutakis A (2005) Semantic similarity measures in mesh ontology and their application to information retrieval on medline. Master's thesis, Technical University of Crete, Greek, (2005)

http://en.wikipedia.org/wiki/SPARQL

http://www.dodenterprisearchitecture.org/Pages/default.aspx

http://en.wikipedia.org/wiki/Resources,_events,_agents_(accounting_model)

http://en.wikipedia.org/wiki/Siri_(software)

Ionita C (2008) Building domain specific languages for voice recognition applications Revista Informatica Economică nr. 2(46)/2008

Li Y, Zhai J, Chen Y (2005) Using ontology to achieve the semantic integration of the intelligent transport system. In: Proceedings of 2005 international conference on management science and engineering (12th), (Vol III). Vienna, Austria, pp 2528–2532

Lowe H, Barnett G (1994) Understanding and using the medical subject headings (mesh) vocabulary to perform literature searches. JAMA 271:1103–1108

Massie T, Obrst L, Wijesekera D (2008) TVIS: tactical voice interaction services for dismounted urban operations. The MITRE Corporation, George Mason University

Meehan TF, Masci AM et al (2011) Logical development of the cell ontology. BMC Bioinformatics 12(1):6

Parry DT (2005) Fuzzy ontology and intelligent systems for discovery of useful medical information. Ph.D. thesis, Auckland University of Technology

Rodger JA, George JA (2010) Adapting the task-technology-fit model and smart data to validate end-user acceptance of the Voice Activated Medical Tracking Application (VAMTA). In: Neustein A Ph.D. (ed) Advances in speech recognition: mobile environments, call centers and clinics. Springer Science + Business Media, LLC, New York/Heidelberg

Rodger JA, Pendharkar PC (2007) A field study of database communication issues peculiar to users of a voice activated medical tracking application. Decis Support Syst 43(2):168–180

Scharenborg O (2007) Reaching over the gap: a review of efforts to link human and automatic speech recognition research. Speech Commun 49(5):336–347

Schorlemmer M, Kalfoglou Y (2008) Institutionalising ontology-based semanticintegration. Appl Ontology 3:131–150

Siniscalchi M, Lee CH (2009) A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition. Speech Commun 51(11):1139–1153

Sleeman D, Ajit S, Fowler DW, Knott D (2008) The role of ontologies in creating and maintaining corporate knowledge: a case study from the aero industry. Appl Ontology 3:151–172

Trappey AJC, Trappey CV, Hsu F, Hsiao DW (2009) A fuzzy ontological knowledge document clustering methodology. IEEE Trans Syst Man Cybern B Cybern 39(3):806–814

Westra R (2002) Marxian economic theory and an ontology of socialism: a Japanese intervention. Capital and Class 78:61–85

Wisnosky DE (2012) Bringing it all together! DoD enterprise architecture conference, Miami Florida, April 2012

Zadeh LA, Kacprzyk J (1992) Fuzzy logic for the management of uncertainty. Wiley, New York

Chapter 13
Revisiting TTS: New Directions for Better Synthesis

Jonathan G. Secora Pearl

Abstract It is sometimes opined that TTS is already good enough. This chapter argues that more work still remains in order to achieve the promise of humanlike speech synthesis and presents a case for expanding the field to include a broader interdisciplinary approach, thereby setting the stage for a future where synthesized speech may be as natural as human communication.

Introduction

Is text-to-speech (TTS) already good enough? Many in the industry are resigned to that view and believe that our focus should shift away from further work on speech synthesis and toward improving speech recognition and natural language understanding. In a recent talk, Kim Silverman, Manager of Spoken Language Technologies at Apple Computer Corporation, described the "long-standing, kind of friendly rivalry and banter between people who do research in speech recognition and people who do research in speech synthesis" noting that the former contend "speech synthesis is a solved problem."[1] Turf wars aside, this view begs the retort: In what way is today's speech synthesis resolved? If TTS is good enough, just what is it good enough for? Is there anyone who would argue that the technology is poised to replace voice-over actors in animated films; ready to read their children bedtime stories; or able to port a voice into a foreign language, dialect or accent for localization

[1] Kim Silverman in an April 16, 2012 talk at the International Computer Science Institute (ICSI) at the University of California—Berkeley. Author's transcription. [http://www.youtube.com/watch?v=zBozX97IxFk]

J.G.S. Pearl, Ph.D. (✉)
Perceptral, 403 6th Street, Racine, WI 53403, USA
e-mail: jonathan.pearl@perceptral.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_13,
© Springer Science+Business Media New York 2013

of business or entertainment media? The overarching question remains: What do we wish to accomplish with speech technology? If the world is our canvas, and our imagination the only hurdle, what will tomorrow sound like?

At every stage of technology, the status quo by definition is good enough at least to maintain the status quo. Innovations are not driven by satisfaction with already-assured outcomes, but by dreams of what is better. Envision with me a future for speech technology wherein human computer interactions (HCI) are indistinguishable from communication between humans; in which synthesized voices assume different accents and registers in response to the communicative environment; where those same voices learn to speak multiple languages with native or near native pronunciation; and where they can do all those things while retaining the timbre and voice quality of a live human-being throughout.

Consider a world where audiences can hear voice reconstructions of historical figures, such as Abraham Lincoln's own voice declaiming the Gettysburg Address and where dubbing of films, television shows, games and informational broadcasts can retain the voice of the original actor or speaker. Imagine a time when anyone can wield a personalized synthetic voice that sounds just like them yet is able to communicate autonomously, reading documents to others; speak a foreign language that they, themselves, do not know; and translate their words in real time and in their own voice to friends, colleagues, and strangers anywhere they go.

What technological advances are required to transform these dreams into realities? While required advances in artificial intelligence (AI) and machine foreign language translation (MFLT) go beyond the scope of this paper, the capabilities of sound-generating systems will need to proceed apace, in anticipation of future advances in MFLT and other related fields. Speech synthesis technologies must also outstrip present standards rather than remaining enthralled to them, and adopt encoding and notation systems that are richer and more advanced than those that exist today. Either we innovate or stagnate. The choice is ours.

Background

Before we embark on the journey to tomorrow, let us consider where we are today and how we got here. Speech synthesis as we know it has been around since the early part of the twentieth century. Initially, mechanical wizardry attempted to imitate vocal production itself. A mechanical source reproduced the vibrations of the human glottis, and various mechanisms replaced the vocal tract, tongue, teeth, and lips. An alternative approach soon emerged to focus, not on the mechanics of sound production, but on the results of that activity. As a result of this shift, the formant synthesis model emerged. It is predicated on the principle that differential patterns of acoustic vibrations evoke the impression of particular speech sounds which may be interpreted by a listener in predictable ways.

Natural sound is characterized by a fundamental frequency which is enriched by a series of subsidiary vibrations that correspond to multiples of the fundamental.

A simple example is a guitar string. When plucked, the string vibrates as a whole once per period, while each half of the wire vibrates twice as fast; each third, three times as fast; and so on. These component vibrations are known as partials, harmonics or overtones. Speech sounds operate in similar fashion. The cycling of the glottis is comparable to the guitar string: it provides the fundamental vibration. The vocal tract serves as a filter on the vibrations, much like the musical instrument.

Formants are frequency ranges in which amplitude is increased relative to other regions. Differences in the shape and material of a sound-producing mechanism produce different formant patterns. These patterns of relative amplitude prominences in the partials are perceived as the unique timbres of each musical instrument or linguistic sound. Formant patterns distinguish a clarinet from a violin or, in the case of speech, between an /o/ in "boat" and an /æ/ in "bat". Exploiting this principle, artificially-produced audio can be perceived as linguistically-meaningful sounds.

As recording technologies advanced, concatenative speech synthesis became dominant. Concatenative speech synthesis is a method whereby soundstreams of recorded speech are segmented into units, labeled, selected, rearranged and adjusted to provide recognizable speech as a chain of linked units. One well-known, early example of this approach is the speaking clock that was created a couple generations ago. It would say, "At the tone the time will be… six… thirty-six … and ten seconds." The most natural-sounding, speech synthesis we have in common use today is based on this type of concatenation. The principal differences between concatenative systems are the segments or units selected, how they are stored and modified, and the means by which they are combined. But, those details are anything but trivial.

State of the Art

One notable, recent effort by Acapela Group, done in collaboration with AssistiveWare, has created arguably the best authentic children's TTS voices to date.[2] "Rosie" and "Harry" (British English) and "Josh" and "Ella" (American English) raise the bar for applications that give voice to children otherwise unable to communicate verbally. Until now, most children's TTS voices were modified adult voices, altered unsatisfactorily in an attempt to make them sound youthful. These are amazing and potentially life-changing products. Notable as they are, they share many of the limitations of other synthesis technologies in use today, as we will explore. They are not proof that we have reached the pinnacle of advancement in speech synthesis, only validation that we are on the right path.

To illuminate what those limitations are, let us consider the description of the speech synthesis process given by Vincent Pagel, Acapela Group's Head of Research

[2] http://www.assistiveware.com/groundbreaking-project-two-authentic-british-childrens-voices and http://www.assistiveware.com/making-new-american-childrens-voices-josh-and-ella.

& Development. [3] He presents the example of the word "impressive" being made up of chunks from "impossible", "president", and "detective". The result is quite impressive, natural sounding, and in a child's voice. The problem is that, as with other contemporary speech synthesis, the production is entirely static. Each instance of "impressive" is an exact duplicate of the last. The relevant syllables of similar words (impression, expressive) would be built with identical component parts. Such carbon-copy duplication, when appearing in a phrase like "my impression of the performance was expressive, really impressive" would jar the listener rather than serve, through slightly varied pronunciation of the repeated syllables, to highlight the alliterative gesture.

There is not a synthetic-voice technology on the market today that can truly claim to sound human, more than once. The algorithms that produce "impressive" or "turn right on Apple Orchard Way" never vary in the way they select and arrange the audio elements that make up the sounds and words. But humans rarely, if ever, repeat the same thing exactly the same way. Humans rarely, if ever, repeat the same thing exactly the same way, unless they wish to emphasize a point. Why should computers?

When you use standard approaches the principal means to create a different intonation, shift emphasis to another syllable, or vary pronunciation is to select and combine entirely different segments. That is a rather inefficient process. The segments of sound upon which these voices are built remain indivisible black boxes, though a sufficiently large database can create the illusion of flexibility. Such flexibility is useful, when high, middle, and low pitches are sought for a given word. This may be needed to provide variety in presentation relative to the placement of a word in a phrase or sentence. A database may include three exemplars of the word "left" to be used at the beginning, middle, or end of a clause. But each time the same word appears in a similar location, its utterance will sound exactly the same. The size and nature of the segments or units used becomes the limiting factor.

In general, current technology employs phoneme-delimited segments. These commonly take the form of diphones—which contain parts of two phonemes and the transition between them—or that of phoneme clusters, such as syllables. The theory behind diphones is that the midpoint of phonemes is more stable than the transitions between them. Consequently, attaching different units at phoneme midpoints provides a smoother sound than might be accomplished by combining whole phoneme units with each other. The principle behind using larger segments is that they require fewer seams between units, reducing the number of joints that the system must contend with. The focus, therefore, is on selecting the best match between segments to create the desired sequence of sounds.[4]

One creative approach using concatenative technology has been applied to the pronunciation of foreign words. This approach is called "phone mapping." It aligns the phonemes that would normally be assigned to a word in its native language with their closest representations in the database of speech sounds in

[3] http://www.youtube.com/watch?feature=player_detailpage&v=M2oVchJIC2o.

[4] Black, Alan & Kevin Lenzo (2001).

another language using a single voice for sequences in both languages. The result is that non-native words may be spoken, but only with a foreign accent. This occurs because differences in pronunciation often go beyond standard phoneme labels. Aspirated and unaspirated consonants, for instance, are not normally assigned different phoneme labels, because these differences principally appear at a subphonemic level. Furthermore, the sound rules in a language may not consider these variations categorically different which means they do not warrant different representations (aspirated and unaspirated) in the database. Phone mapping at the phoneme level or above will necessarily miss these language-specific differences, meaning that a voice ported to a new language will carry with it the accent of its source language.

To illustrate this point, American English distinguishes /p/ and /b/ in large part by the duration and degree of aspiration, whereas standard Finnish lacks this distinction.[5] While /b/ exists in borrowed words in Finnish, /p/ is typically unaspirated in pronunciation. Thus the two sounds are most likely heard by Finns as variants (called "allophones") of the same phoneme, rather than as categorically distinct phenomena.[6] The plosive and aspirate subcomponents of these sounds are identifiable, distinct, and could readily be stored as separate items in a database but only at the sub-phonemic level. At present, however, they are not. That forecloses the facility to combine or segregate them in novel or atypical configurations.

It is not that speakers from one language or dialect cannot produce the sounds that are native to another language. We know they can, because some speakers produce near-native pronunciation in a second language. Even though the superconfigurations may not be native to their speech they can do this because many of the necessary subcomponents can be found elsewhere in the sounds they produce. It is only in the case of rare and unusual sounds, such as the clicks of !Kung or Xhosa, that equivalents may not be found in other languages. Aspirations or their equivalents exist in Finnish speech, just not in locations or combinations typical of those in American English. There is little reason a synthetic voice could not be modified to mimic or produce the sounds desired. For instance, a synthetic Austrian voice could become fluent in Swedish, if so desired. All that is needed to travel between languages and dialects is a finer degree of analysis and measurement than existing phoneme-delimited systems permit.

[5] Frequently, the voiced/voiceless distinction between /p/ and /b/ is described in terms of voice-onset time (VOT), specifically that shorter voice onset times are related to the voiced bilabial plosive /b/ , and longer VOT is associated with the voiceless variant /p/. But while voicing is a useful descriptor of speech production, namely the engagement of glottal vibrations, it is not a terribly useful description of the acoustic signal. In our own research aspiration is a more reliable indicator of the /p/-/b/ distinction.

[6] Without aspiration, the VOT of /p/ is necessarily shorter, thus reducing any possible productive difference between them.

Flawed Assumptions

Fred Jelinek, the long-time director of speech technologies at IBM and a pioneer of speech recognition and natural language processing, is often cited as a source for the view that having linguists on staff is counter-productive to the effectiveness of a system. Unfortunately, what may have begun as an offhand observation in the 1980s relative to a particular group of researchers dealing with a specific problem, has turned into a license for ghettoization of the entire field. Because of that, the overwhelming majority of participants are trained in a diminishingly small set of disciplines almost exclusively computer science and electrical engineering. While credentials used by human-resources departments and hiring managers are well-suited to maintaining the status quo, they may not always lead to innovation.

The consequence of that endemic policy is that, today, the broad and rich domains of human communication and human-computer interaction (HCI) are frequently viewed through myopic lenses. With an unduly limited-view of the tasks at hand, and too constrained a toolset, there is a tendency to over-restrict the problem space, identifying solutions too narrowly. As the primatologist Frans de Waal once put it: "We cannot afford to look through a single pair of glasses; we need lots of different glasses to see reality."[7] Today's teams are increasingly expert at signal processing. In order to create synthetic voices that are perceived as human-like we also need to understand how humans perceive speech.

Humans are adept at recognizing and identifying differences in pronunciation, accent, timing, emphasis, and nuances of intonation and micromodulation in ways that standard transcription systems cannot begin to capture. The fault lies in the imprecision that those systems evince. What might best serve would be expanding expertise to foster innovation in transcription and encoding standards. In many cases, however, major developers have simply thrown up their hands rather than exit established comfort zones. Witness the view from one of IBM's leading speech scientists, Andy Aaron, in a March 7, 2009 interview on National Public Radio:

> [T]he idea is, we've given up at trying to model the human speech production system. It's too complicated. There are too many parameters. Nobody understands it. So instead, what we'll do is we'll audition lots and lots of voice actors, pick one that we like. And record them reading untold amounts of sentences, over a period of maybe a month, until they're exhausted [laugh] and then we, uh, take those sentences, chop them up into individual pieces, called phonemes, and build a library of that person's voice.[8]

The problem is not that speech production is too complicated or even that its process is poorly understood. It is rather that research has been focused on a narrow swath of inquiry at the exclusion of all others.

The phoneme is typically described as the smallest, contrastive unit within the sound system of a language. That characterization is somewhat misleading. Whatever they are, phonemes are not sounds. No phoneme has a one-to-one

[7] de Waal, Frans (2001).

[8] Author's transcription. [http://www.npr.org/templates/story/story.php?storyId=101585667].

mapping where labels unambiguously correspond to acoustic patterns.[9] Rather, the phoneme is a concept—an inconsistent one at that. Sometimes it refers, to functional categories in spoken language. For example, the word "cat" is /k/-/æ/-/t/ regardless of the various allophones that appear in pronunciation. At other times it is a set of typically binary articulatory features (voiced/voiceless, tense/lax, rounded/unrounded, high/middle/back), describing the location or aspect of the articulators (e.g. tongue, lips, glottis). Tellingly, phonemes are sometimes called "the atoms of spoken sound."[10] As physics has shown, the atom is not as indivisible as originally conceived.

In terms of phonemes, there is no distinction between a New Englander's pronunciation of "dog" and a South Texan's. But we can all hear it. It is time that synthetic voices were able to produce it as well. Dictionaries of pronunciation for TTS engines most often ask "how should this text be spoken" without regard to who is doing the speaking.[11] Perhaps the question ought to be "how does this speaker in this context at this time pronounce this text?" Answering the latter question requires a deeper comprehension of speech sounds than phonemic analysis permits. Consider how one might deal with the observation that two speakers, one from Chicago, and one from Dublin, pronounce the word "house" differently. We might say they use different phonemes to pronounce the same word. Or we might say they use different sounds to pronounce the same phonemes. Which description would allow us to take a TTS engine built on the first voice to speak with the accent of the other? That's a trick question, of course, because without new encoding standards, the answer is "neither."

The Inadequacy of Current Standards

Overcoming these limitations will necessitate not only advances in the underlying technology, but improvements in the means by which speech sounds are labeled and described. This is tantamount to "splitting the atom" of the phoneme to examine the quarks and muons of speech that emerge. The latest version of SSML (Speech Synthesis Markup Language) still lacks standards for most prosodic alterations.[12] Syllable-level markup is only partially addressed, and sub-syllabic, phonemic, and subphonemic alterations of prosody remain completely out-of-scope.[13]

---

[9] Tatham, Mark & Katherine Morton (2006): 4, 11, 24, 199–200.

[10] http://www.explainthatstuff.com/how-speech-synthesis-works.html.

[11] http://www2.research.att.com/~ttsweb/tts/faq.php#TechHow.

[12] "Speech Synthesis Markup Language (SSML) Version 1.1: W3C Recommendation 7 September 2010": http://www.w3.org/TR/speech-synthesis11/.

[13] Daniel Burnett, "The Internationalization of the W3C Speech Synthesis Markup Language," SpeechTek 2007 & "Speech Synthesis Markup Language Version 1.1 Requirements, W3C Working Draft 11 June 2007": http://www.w3.org/TR/ssml11reqs/.

The working group for SSML contends that a lingering lack of consensus regarding the definition of a base unit remains the main hurdle to controlling something as fundamental to communication as speech rate. "A primary reason for this was lack of agreement on what units would be used to set the rate—phonemes, syllables, words, etc."[14] This lack of consensus, however, may reflect the reality that none of these units adequately addresses speech prosody. Phoneme-delimited systems are best suited to representing language at the gross level of words, phrases, and sentences. They prove less capable of adapting to the gradations and nuances of everyday speech prosody—nuances that we need to harness in order to produce human-like synthesis.

To exemplify the importance of these unattended aspects of prosody, let us consider a neurological condition described as "foreign accent syndrome" which affects a speaker's ability to produce micro-variations in speech—the very ones that contribute most to a speaker's accent.[15] In a seminal 1947 article, Oslo-based clinician Monrad-Krohn described the case of a 30-year old Norwegian woman who suffered shrapnel damage to the left frontal region of her brain from a wartime bombing. This injury disturbed the natural flow of melody, rhythm, and accent in her speech, which the author dubbed *dysprosody*. Here is how Monrad-Krohn described the issue he confronted:

> In the Scandinavian languages there are two kinds of accents, viz. "one-syllable accent" and "two-syllable-accent" (the pitch gliding down on the same syllable in the latter). Bönder (meaning peasants) has the accent on the first syllable and the d is not sounded. Bönner (meaning beans or prayers) also has the accent on the first syllable. Yet the accent on the first syllable of "bönder" is a one-syllable one; but a two-syllable one in "bönner." A good test for the melody of language can thus be had in Norwegian by asking the patient to say: "Bönder plukker bönner" (meaning "peasants pick beans").[16]

This pronunciation task that he describes is one the patient initially failed. Common transcription standards for prosody, such as ToBI ("Tone and Break Indices"), have no means to describe this sort of phenomenon. [17] Indeed, they perform inadequately in describing a great range of perceptually-meaningful sound-events. ToBI provides no explicit encoding for timing effects (which among other things are crucial turn-taking cues) and greatly underspecifies the details of other features, like subsyllabic pitch movements. ToBI is typical of the transcription systems available for use by developers of speech technologies today, influencing and limiting the applications they can support. The linguist Jane Edwards remarked: "…no transcript is completely theory-neutral or without bias."[18] Without explicitly noting those biases and

[14] Cf. Section 8.8 in "Speech Synthesis Markup Language Version 1.1 Requirements," as cited above.

[15] Mid-twentieth Century researchers having access to some of the first reliable machine measurements of sound made note of the relevance of subsyllabic and subsegmental pitch movement (Cf. Léon and Martin 1970: 32), but attention seems to have faded over the ensuing decades.

[16] Monrad-Krohn, G. H. (1947): 412.

[17] Silverman, Kim, et al. (1992).

[18] Edwards, Jane (1993).

assumptions, developers and end-users of applications subject to those standards are withheld the rich tapestry of speech.

Kim Silverman, who was first author on the papers that established ToBI, remarked to this author in response to a question regarding the absence of markup standards for affective intonation: "If you ask me what are the acoustic parameters of affect, I'll ask you, what are the acoustic parameters of past tense."[19] While some aspects of affect are encoded in non-prosodic elements of speech, as Silverman intimated, (e.g., word choice, manual gesture, and eye contact) a clear majority is evident in prosody—intonation, tone of voice, timing. Indeed, the cadences, the irony of his delivery, and the unmistakably Australian flavor of Silverman's own speech remain completely absent from the transcribed quotation above, and unaddressed in the methods of transcription that he defends.[20]

Despite the importance of subcomponent analysis and microintonation, as noted above, even cutting-edge methods of marking variations in pronunciation are limited by standard, phoneme-based classifications.[21] This is true, whether those differences are between dialects and languages or due to affect or intention. The means by which existing systems segregate similarly-labeled units by accent, for instance, require separate databases for each group of speakers (e.g. UK vs. US English, Finnish vs. Estonian, European vs. South American Spanish).[22] Degrees of similarity between phones from different databases are not considered. This leaves the relationship among the various pronunciations unknown and unknowable since the phoneme label represents the finest degree of detail that can be addressed by these methods. Units of sound in different databases that are assigned the same label are considered effectively equivalent, regardless of measurable differences in their substructure. In essence, the phoneme is presumed to be an atomic unit, when clear evidence suggests it is not, inevitably limiting accuracy, precision, and flexibility.[23] The absence of finer-grained analysis is problematic because subphonemic and microprosodic structures, which are essential characteristics of human speech, are undetectable by these means, and therefore unavailable to us for control under speech synthesis.

Unfortunately, a long-term trend has been in the opposite direction, away from smaller more precisely defined units, and toward larger and longer segments of sound. A frequently unchallenged view posits that only large-scale, suprasegmental events are worthy of measurement and classification.[24] It is our contention that this attitude is unwarranted and counterproductive. Rather, the key to unlocking prosody at a macro scale is understanding it in the micro scale. This requires the establishment

[19] Private telephone conversation, 2007.

[20] While some of these details might be recorded in meta-data appended to the transcript, they are not in any way formalized or standardized as part of the transcription itself.

[21] Kadirkamanathan, M. & C. J. Waple (2011), *Automatic Spoken Language Identification Based on Phoneme Sequence Patterns*, US Patent Application 2011/0035219, §[0025].

[22] Cf. Kadirkamanathan & Waple (2011), §§ [0004], [0013].

[23] Tatham, M & K. Morton (2006).

[24] Lehiste, Ilse & Gordon Peterson (1961); and Patel, Aniruddh (2008).

of standards for describing and comparing local phenomena that have largely been ignored.[25] Microintonation provides precisely the sort of data we require to give a synthetic speaker personality and accent, emotion and intentionality. Without the capability to modify sound at this level, and without standards for encoding them, this task remains impossible.

Contemporary speech synthesis has reached a plateau in its development and requires a shift in our thinking if we are to reach the next summit. We seem to have forgotten that among the fundamental and salient elements of speech is the variety of sound itself: the subtle gradations of pronunciation and emphasis; the nuances of intonation and prosody. It does not really matter how many synonyms a TTS engine can use for a word or paraphrases for a concept, if each one reminds listeners that they are interacting with a machine. Creating animated voices may require a leap equivalent to exchanging binoculars for microscopes: abandoning our enchantment with phonemes and suprasegmentals to delve deeper into subphonemic relationships and microintonation.

A Way Forward

As I hope has become clear throughout, numerous fields of study from cognitive neuroscience to music perception to the psychology of language acquisition have valuable contributions to make to speech technology, if only we would let them in Pearl, Jonathan (2005). In addition to work on Foreign Accent Syndrome cited above, a wealth of literature has considered aprosodias, typically associated with brain regions of the right hemisphere homologous to those in the left hemisphere that are deemed crucial to language (e.g. Broca's and Wernicke's areas). These impair a person's ability to comprehend or produce the affective and sometimes lexical/grammatical aspects of speech prosody, whether or not their phonemic awareness or generation has been altered.[26] Studies of dyslexia and aphasia are also potential drivers of novel approaches to speech technology. Language learning, in which not only vocabulary and grammar are acquired, but accent, pronunciation, timing and intonation are key elements, is both a source and a market for dramatic improvements to our capabilities. Music performance-studies offer another opportunity for fruitful cross-fertilization and collaboration. Evidence shows, for instance, that musicians routinely speed up at the beginning of phrases and slow down towards the end in ways that mirror human performance in speech.[27] But none of this research has yet made its way into the mainstream of contemporary efforts to produce human-like speech synthesis. Until it does, we will likely continue to muddle through,

[25] Hermes (2006) citing House (1990) writes of the "perceptual irrelevance of microintonation". Yet, Hermes himself acknowledges that although microintonation may be insignificant in terms of higher order pitch contours, it retains perceptual relevance for segment classification (p. 32).

[26] Ross, Elliott D. et al, various.

[27] Pearl, Jonathan (2003) and (2007).

failing to ignite the passion of a resigned public. The future of TTS depends on our willingness to discard our attachment to long standing assumptions, to revisit avenues long abandoned, and to mark new trails between fields that may lead inevitably toward innovations tomorrow.

References

Black A, Lenzo K (2001) Optimal data selection for unit selection synthesis. From the 4th ISCA worskop on speech synthesis. Online: http://www-2.cs.cmu.edu/~awb/papers/ISCA01/select/select.html

Bolinger D (ed) (1972) Intonation. Penguin Books, Baltimore

Burnett D (2007) The internationalization of the W3C speech synthesis markup language. SpeechTek. http://conferences.infotoday.com/documents/27/C102_Burnett.pps

de Waal F (2001) The ape and the sushi master: cultural reflections of a primatologist. Basic Books, New York

Edwards J (1993) Principles and contrasting systems of discourse transcription. In: Edwards JA, Lampert MD (eds) Talking data: transcription and coding in discourse research. Lawrence Erlbaum Associates, Hillsdale

Hermes D (2006) Stylization of pitch contours. In: Sudhoff S (ed) Methods in empirical prosody research. Walter de Gruyter, Berlin/New York, pp 29–61

House D (1990) Tonal perception in speech. Lund University Press, Lund

Kadirkamanathan M, Waple CJ (2011) Automatic spoken language identification based on phoneme sequence patterns. US Patent Application 2011/0035219, Assignee, Autonomy Corporation, published 10 Feb 2011

Lehiste I, Peterson G (1961) Some basic considerations in the analysis of intonation. J Acoustical Soc Am 33(4). Reprinted in Bolinger (1972): 367–384

Léon P, Martin P (1970) Prolégomènes à l'étude des structure intonatives. Marcel Didier, Montréal/Paris/Bruxelles, Translated by Susan Husserl-Kapit and reprinted in Bolinger (1972), pp. 30–47

Monrad-Krohn GH (1947) Dysprosody or altered 'melody of language'. Brain 70:405–415

Patel A (2008) Music, language, and the brain. Oxford University Press, Oxford

Pearl J (2003) Hypothetical Universe: a functionalist critique of Lerdahl-Jackendoff. Presented at the Society for Music Perception & Cognition (SMPC) conference, Las Vegas, Nevada. [http://www.musiclanguage.net/conferences/presentations/hypothetical-universe/]

Pearl J (2005) The music of language: the notebooks of Leoš Janáček. Dissertation, University of California, Santa Barbara. [http://www.musiclanguage.net/files/dissertation.pdf]

Pearl J (2007) Varieties of Czech Prosody. In: Meeting of the German Society of Linguistics (DGfS) in Siegen. [http://www.musiclanguage.net/conferences/presentations/dgfs-2007-varieties-of-czech-prosody-presentation/]

Ross ED (1981) The aprosodias: functional-anatomic organization of the affective components of language in the right hemisphere. Arch Neurol 38:561–569

Ross ED (1984) Right hemisphere's role in language, affective behavior and emotion. Trends Neurosci 7(9):342–346

Ross ED, Mesulam M-M (1979) Dominant language functions of the right hemisphere? Prosody and emotional gesturing. Arch Neurol 36:144–148

Ross ED, Edmondson JA, Burton Seibert G (1986) The effect of affect of various acoustic measures of prosody in tone and non-tone languages: a comparison based on computer analysis of voice. J Phon 14:283–302

Ross ED, Edmondson JA, Burton Seibert G, Homan RW (1988) Acoustic analysis of affective prosody during right-sided wada test: a within-subjects verification of the right hemisphere's role in language. Brain Lang 33:128–145

Ross ED, Anderson B, Morgan-Fisher A (1989) Crossed aprosodia in strongly dextral patients. Arch Neurol 46(Feb 1989):206–209

Silverman K, Beckman M, Pitrelli J, Ostendorf M, Wightman C, Price P, Pierrehumbert J, Hirschberg J (1992) TOBI: a standard for labeling english prosody. In: Proceedings of the international conference on spoken language processing (ICSLP-92). Edmonton, Alberta, Canada, pp 867–870

Tatham M, Morton K (2006) Speech production and perception. Palgrave Macmillan, New York

Chapter 14
"Super-Natural" Language Dialogues: In Search of Integration

Bruce Balentine

Abstract By definition, the goal of NL speech research has been "natural" human language—that is, faithfully simulating human-human conversation with human-computer dialogues. The word "natural" here is enclosed in quotation marks to emphasize its narrow definition. Attempts to explore alternatives to this narrow definition have attracted little interest. But there are other ways to incorporate speech into effective and intelligent user interfaces. Text-to-speech synthesis, for example, can easily exceed human capabilities—encompassing a wider pitch range, speaking faster or slower, and/or pronouncing tongue-twisters. Similarly, non-speech audio can provide prompts and feedback more quickly than speech, and can also exploit musical syntax and semantics. Speech recognition dialogues, in turn, can master pidgin languages, shortcut expressions, cockney-like underground tongues, and non-speech sounds. User interfaces designed around these super-human technology capabilities are known collectively as *super-natural* language dialogues (SNLD). Such interfaces are inherently multimodal, and their study is inherently interdisciplinary.

Introduction

In this paper, I describe research and prototyping underway at several laboratories, providing a background on more than three decades of work in music, non-speech audio, multimodal language and gesture that I believe has been under-explored by the natural speech and language communities. I then go on to propose an SNLD

B. Balentine, M.Mus. (✉)
EIG Labs, Enterprise Integration Group E.I.G. AG,
Weinbergstrasse 68, Zürich 8006, Switzerland
e-mail: bruce@eiglabs.com

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3_14,
© Springer Science+Business Media New York 2013

working environment based on *soundspace*, a multimedia illusion created through audio cues. Finally, I make the argument that all of the disciplines required to achieve a standard and integrated 4D super-natural interface are fully mature.

The value of the SNLD approach is that it proposes a model that is orthogonal to certain NL assumptions (Christian 2011) that have dominated speech research and product development for decades. By explicitly redefining "natural" goals, all of the arguments and obstacles that stand in the way of alternative designs become moot. These arguments and obstacles generally conform to two archetypal quotes:

- "That's not how humans do it"; and/or,
- "That's not natural".

By taking a fresh look at the problem, talking points can be devised according to the merits of alternatives without reference to the principles of human and "natural" conversation: a machine need not use speech modalities the way humans do (Balentine 2007a). Whether this specific paradigm eventually finds its way into any product is not so important. We will use it here as a reference and a reflector—asking and answering questions about usability, feasibility, desirability, and value.

It is well-known that a natural language speech application "resembles a command-line interface in that it hides the application's functionality" (Martin et al. 1996). This fact is viewed as a feature by many HCI practitioners, but it also causes challenges for the user, among them:

- Knowing what to say;
- Mastering error-recovery protocols;
- Discovering effective multimodal paradigms;
- Coping with data presentation; and,
- Interacting with auditory machine output.

I and my colleagues at Enterprise Integration Group (EIG) have been studying these challenges for more than two decades. Despite technology improvements on the input (ASR) side of the equation, methods for closing the conversational loop with effective machine output continue to be demanding (Balentine et al. 2000).

Assuming that there is a role for SNLD on the HCI palette, how might one go about searching for design techniques? The obvious answer is to look to the most successful and ubiquitous user interface paradigm to date—the WIMP interface—and then to update it with new sonification techniques that bring the dimension of time into the existing WIMP standard model.

Background on WIMP and Sonification

If the graphical user interface (GUI) represents the standard HCI model for computer interfaces, then the so-called WIMP (windows, icons, menus, pointers) is clearly its standard-bearer. First conceived at Xerox PARC by Alan Kay, Larry

Tesler, Dan Ingalls and a number of other researchers based on earlier work by Douglas Engelbart, who invented the mouse at SRI in 1964, the WIMP interface now incorporates standardized user behaviors that are known to most of the population of the world.

No such bounded set of interactive designs has ever enjoyed the popular success that the WIMP interface has achieved. A small set of common reusable devices—known as widgets—are applied by designers to give a user access to almost any kind of computing tool. Widgets include windows, menus, list managers (a kind of menu), and pushbuttons. The ability of early WIMP designers to generalize their widgets is remarkable when you consider that most of today's applications and task domains didn't even exist at the time the widgets were designed. Kay and his team—rather than exhaustively analyzing specific task domains—found broad general abstractions that they surmised could service a wide array of different tasks.

Sonification is the broad term for the field of study that investigates human factors of auditory media (Kramer et al. 2010). Sonification is concerned primarily with non-speech audio, but also includes a number of speech technologies, especially text-to-speech (TTS) synthesis. The field has flourished in recent years, and has a long history of effective R&D.

In the following few pages, I will give an overview of WIMP widgets and techniques for sonifying them—primarily to introduce terminology, and to show their importance in SNLD designs.

Windows

What constitutes "naturalness" in a WIMP interface is driven by the laws of physics. When we use the word "intuitive" in our discussions, what we mean is the degree to which an interface conforms to our assumptions and prior experience about how the physical world works. In HCI terms, the window is perhaps the most interesting and complex WIMP device, so we'll use it as a tool for understanding this principle.

A window is sometimes described as a *container* into which one can deposit data. A better way of thinking, however, conceptualizes the body of data as one physical object in the background—a *document*—with the window as a second foreground object providing a viewing and manipulation port that gives the user access to the data.

There are a couple of ways to visualize this manipulation. When I operate the scroll box or thumb in Fig. 14.1c, the window remains fixed, but I may imagine that I am turning some unseen scrolling mechanism (d). Pushing upward turns the scrolls in one direction, winding the document *down* from the top scroll to the bottom. Reversing direction reverses the scrolling, and the parchment moves *upward* from the bottom scroll to the top—thereby exposing information later in the document.

If I move the pointer into the document itself, I may get a *hand* (e)—a different icon with which I can manipulate the document directly. In this visualization, my directions are reversed. I can *grab* the document and *pull it down* to view information earlier in the document. Or I may *push the document up* to expose information later in the document.

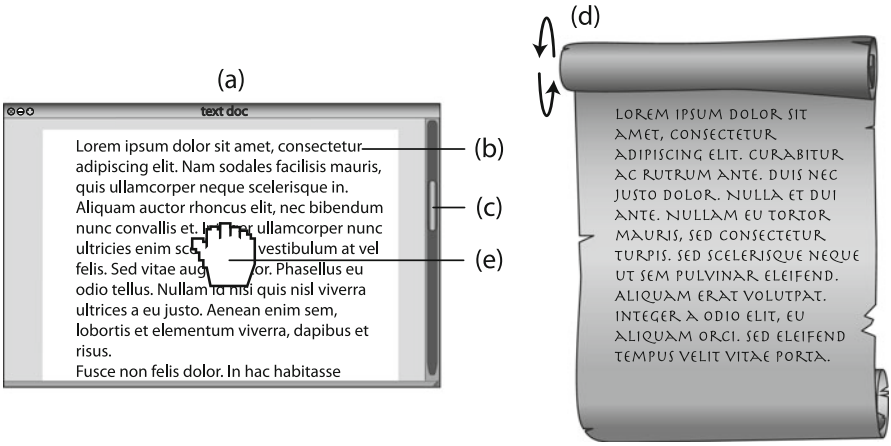

Fig. 14.1 Window (**a**) provides a view into a virtual document (**b**); the scroll box or thumb (**c**) indirectly manipulates scrolling machinery (**d**), while the hand (**e**) allows direct manipulation of the virtual document

Switching back and forth between moving the scroll box and moving the hand to accomplish the same action creates little user confusion. This lack of confusion is quite remarkable when you consider that you must move the scroll box *in the opposite direction* as the hand to accomplish the same result. Both actions correspond to one set of physical laws, and those laws are consistent between actions.

A Sonic Window

The auditory version of a window is shown in Fig. 14.2. Any spoken audio of any duration may be placed into this window. The dimension of interest is now time instead of space.

The time-grid (Fig. 14.2a) presents 10 equidistant blips for each of the three sections. The *tempo* of the grid therefore corresponds to the *size of the thumb* in a GUI menu—conveying the overall duration of the audio through a parallel audio channel while the user listens concurrently to the message (b). The message is superimposed (mixed) with the grid. Section markers (c) use musical syntax to show progress in the form of a chord sequence (I IV V I). The authentic cadence aligns with the end of the message (d), indicating completion or closure.

As shown in Fig. 14.3, the user manipulates the window by speaking. Command words allow major jumps to the beginning (a), middle (b), or end section (c), as well as fine-navigation with commands such as “back up” (d) or “faster/slower” (not shown). The user speaks these commands while concurrently listening “through” the musical grid to the audio message. The musical grid constitutes the foreground “window” and the audio message becomes the “background document.”

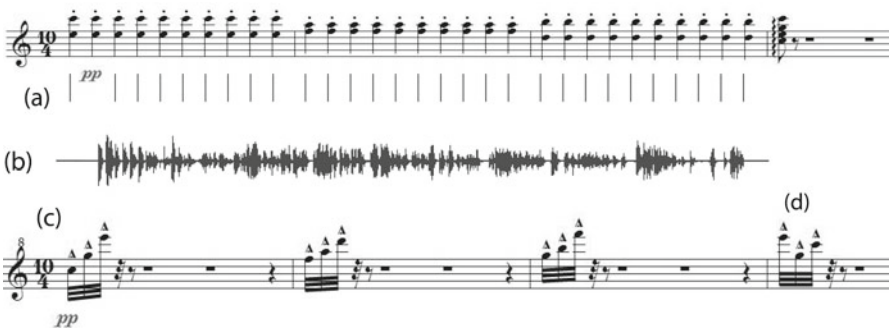

Fig. 14.2 Sonic window provides interactive scrolling capability in the dimension of time as opposed to space. The time-grid (**a**) presents 10 equidistant blips for each of the three sections. The tempo of the grid therefore corresponds to the size of the thumb in a GUI menu

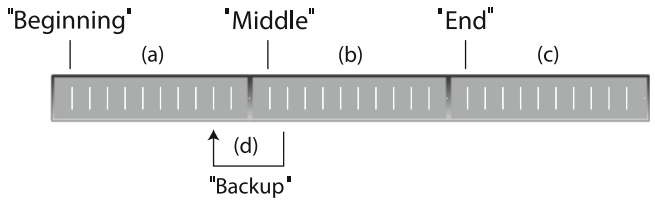

Fig. 14.3 Command words give the user voice control of the sonic window. Sections (**a**, **b**, **c**) are easier to conceptualize than absolute time in seconds, and the ability to replay a sequence that has just gone by (**d**) solves a common listening problem. The command words are easy to remember, and remain the same for all types of spoken audio whether recorded or synthesized

There are a number of other sonic window details—beyond the scope of this overview paper—that make the sonic window broadly useful for a number of tasks. One important mention must go to the "zoom" feature. Any of the three sections may be enlarged through a zoom command, leading to a newly-calculated grid (with a 3x tempo) and a modulation of musical key. Extremely large audio streams may thus be traversed in sections, with multiple zooms moving through the circle of fifths. The feature is important in proofreading applications, wherein a text document is read aloud with a text-to-speech synthesizer. Nested zooms support such applications, as well as screen-reading, voice-note skimming,[1] and similar complex traversals of large spoken documents.

Martin Mull once said, "Writing about music is like dancing about architecture," so I must note here that written descriptions of sonic designs always sound more complex than they are in use. Rather than write even more words to do only

[1] Audio searching and skimming designs have been well-studied. See (Arons 1994) for one of the earliest.

a poorer job of communicating, I encourage the reader to listen to the audio examples accompanying this paper to understand these designs.

Any spoken message can be deposited into a sonic window, making it as universal a widget for temporal manipulation as is the WIMP window for textual and other spatial data. Note the parallels: the WIMP scroll box, scroll bar, and associated machinery are independent of the content. A window does not know or care what the information in the document might be. The machinery of the window is designed to give the user a direct manipulation interface that supports any document, regardless of length, language, or format.

Similarly, the user does not speak about the audio content itself, the user speaks to the sonic window about *locating information within the document*, with commands that give the user direct control over audio navigation. Mastery of the sonic window thus gives the user mastery over time.

Music and Music-Like Sounds

The sonic window relies on a number of musical attributes to provide an intuitive and self-documenting manipulation device. Here, the word "intuitive" is used in exact correspondence to its use in WIMP—it conforms to the laws of physics—in this case, laws that derive from our biological (evolutionary) heritage first as listening organisms, and then later, as our listening skills advanced, to include semantic and syntactic processing of sounds.

Associating musical sounds with meaning depends on a multimodal mapping in which changes to the sound(s) are perceptually correlated with changes in other properties of the user interface. In Table 14.1, (Koelsch 2012) provides a spectrum across which musical sound and meaning can be categorized.

Musical references to the extra-musical world can be *iconic* (the music imitates the sound or quality of an object, e.g., flutes and clarinets mimicking birdsong), *indexical* (music portrays a psychological state, e.g., a somber viola solo portrays loneliness and grief), or *symbolic* (association by convention, e.g., a well-known national anthem melody symbolizes a country). The sonic window grid has an iconic meaning—the ticks or blips are reminiscent of a clock ticking, and thus symbolize the passage of time.

The meaning of tonality in music is what Koelsch would call intra-musical—that is, a principle that is intrinsic to music without need for external reference or symbolization. In the sonic window, for example, the first section of the message marks its transition to the second by changing the implied harmony from tonic to subdominant. The subdominant in turn progresses to the dominant for the final section of the message. The progression sets up an *authentic cadence*, which announces the final conclusion. This harmonic progression "makes sense" to the user without need for explanation.[2]

[2] "Musical sense" is often not consciously-recognized but can still be measured through imaging, observed fluctuations in arousal, or other physiological experiments.

Table 14.1 Categories of musical meaning (Koelsch 2012)

Extra-musical			Intra	Musicogenic		
Iconic	Indexical	Symbolic	musical	Physical	Emotional	Personal

The intra-musical meaning "emerges from one musical element (or a group of musical elements) referencing to another …" (Koelsch 2012).

Musicogenic meaning arises from physical, emotional, and/or personality-related responses. According to Koelsch:

> With regard to emotional musicogenic meaning (i.e., meaning due to feelings evoked in a listener), it is important that music can invoke feeling sensations which, before they are denominated with words, bear greater inter-individual correspondence than the words that an individual uses to describe these sensations. In this sense, music has the advantage of defining a sensation without this definition being biased by the use of words. I refer to this musicogenic meaning quality as "a priori musical meaning".

This distinction between physical/emotional and personal meaning in music is reminiscent of Meyer's *absolute* versus *referential* classification (Meyer 1956).[3]

Menus

Menus of various kinds are ubiquitous in the WIMP interface. These include not only pull-down and pop-up devices that are well-recognized as menus, but graphical devices such as toolbars, ribbons, and galleries. In all cases, the device presents to the user a bounded set of options that are available in the current state, and the user simply chooses from among them.

Various embodiments of the simple notion of a menu are shown in Fig. 14.4. Text menus (a, b) most-closely resemble their restaurant progenitors, while graphical menus (c through e) are more abstract descendants of the same concept. See (Balentine 1999) for a point-by-point comparison of WIMP versus spoken menu components.

Speech Menus

Menus are the best-understood of all artificial speech dialogues, and have been developed, studied, and refined by many practitioners for more than two decades (Lewis 2011). Oddly, design principles that are thoroughly-tested and known to be effective remain unused in most speech products. Indeed, the menu itself is consis-

[3] In a personal communication, Koelsch challenges this statement, arguing that "Meyer's referential/absolute classification is rather reminiscent of extra- and intramusical meaning (terms that Meyer also used)." Koelsch suggests instead, that, "The distinction between indexical extramusical meaning (i.e., the recognition of an emotional expression) and emotional musicogenic meaning (i.e., meaning emerging from the presence of a particular feeling state) is reminiscent of Gabrielson's "perceived" and "felt" emotion."

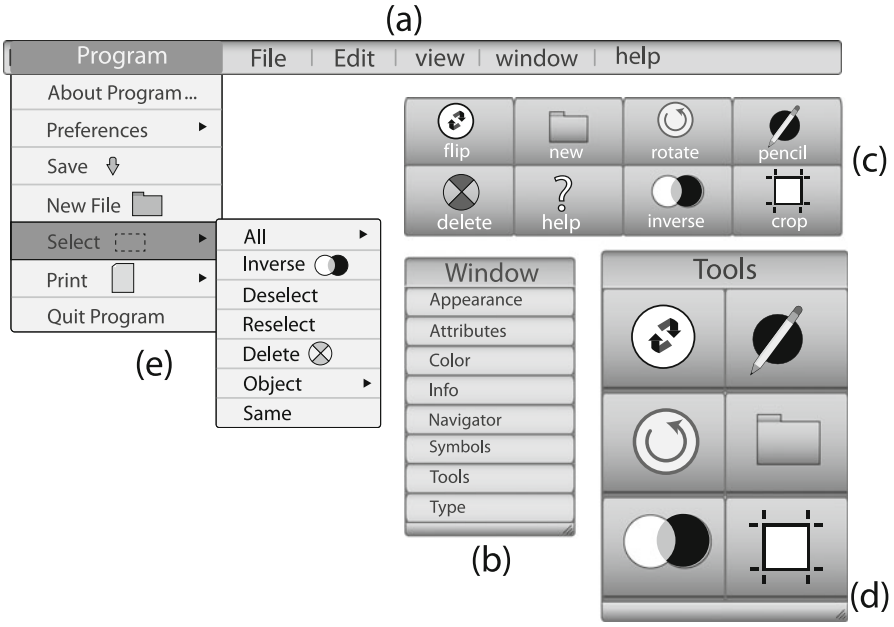

Fig. 14.4 WIMP menus take many forms, and can be textual (**a**, **b**), graphical (**d**), or mixed (**c**, **e**)

tently denigrated by some in the speech community. This is hard to understand given that WIMP relies so extensively on menus, and that menus—even spoken menus—are so demonstrably "intuitive" and "natural" in usability tests.

There are a number of important human factors issues with spoken menus:

- Number of elements;
- Number of words to describe each element;
- Parallelism;
- Pace, rate-of-speech, and pauses;
- Discourse markers; and,
- Error recovery and menu repetition methods.

In addition, multimodal menus (most-commonly touch-tone and ASR) have design challenges that revolve around modality-focus, modality changes, and hidden versus public choices. EIG in particular has done extensive work with parallel prompting, modality stickiness, and turn taking in speech/touch menus.

The maximum and recommended number of elements for a spoken menu has been studied extensively. Commarford et al. (2008) discuss Miller's well-known but often mis-cited guideline, "Seven Plus or Minus Two" (Miller 1956), assessing it against newer models of human memory. Suhm et al. (2001) demonstrate that a "long menu with specific, clearly defined categories can identify the reason for a

call more efficiently than a short menu with broad categories." Their results agree with Commarford's, suggesting that breadth is preferred over depth—that is, more elements in a given menu outperform attempts to shorten that menu by breaking it into two or more smaller menus. Internal EIG research also argues against *under-chunking*, that is, attempting to subsume specific concepts into a smaller number of broader concepts with the goal of shortening menus.

In other words, menu-length tradeoffs are now well-understood: short menus are cognitively easier, but the overall task requires more turns. Long menus are more efficient in terms of turn-efficiency, but must be repeated if they don't work the first time. Similarly, multiple words per list-element sometimes increase menu clarity, but also increase mental-chunking and total menu-length costs. Balentine (2007b) provides some theoretical background on "that's it" versus "hold-in-mind" menus—hypothesizing that the former can contain 20 or more elements while the latter are best kept at or below five—but the hypotheses have not been rigorously examined.

Auditory Icons

In addition to music, recorded natural sounds have been explored and refined by a number of investigators (Serafin et al. 2011). These sound objects are called *auditory icons*.

In Table 14.2, early work by (Gaver 1986) described three categories of auditory icons: nomic, metaphorical and symbolic (top row). Kramer (1994) proposed a broad continuum for classifying such sound objects (second row). Later work (Keller and Stevens 2004) developed a more formal taxonomy (third row). All of the above work has been summarized and explicated by (Walker and Nees 2011) (final row).

At one end of Kramer's continuum, analogic representations correspond to Gaver's nomic category. Nomic or analogic sounds have a direct relationship to the object or concept that they stand for (e.g., the sound of a metal drawer closing represents "closing a file"). At the other end, the correspondence between sound object and referent is more symbolic, abstract or even arbitrary (e.g., the sound of a bulb horn indicates that the printer is out of paper). Located along the middle of the spectrum are signal-referent relations that fall in-between these two poles.

Earcons

The term earcon is usually attributed to Blattner et al. (1989), although a footnote in that paper references (Buxton et al. 1985), an unpublished paper. The concept of an earcon is more complex than that of an auditory icon, in that *families* of earcons can exhibit interrelationships. One auditory icon has a single referent, but earcons in a family participate in a formal organization, providing syntactic meaning that is based on modular, transformational, and hierarchical structures.

Table 14.2 Classification schemes for auditory icons

Gaver	Nomic		Metaphorical		Symbolic
Kramer	Analogic				Symbolic
Keller/Stevens	Direct			Indirect	
	Iconic	Noniconic		Ecological	
Walker/Nees	Direct denotative	Indirect		Metaphorical	Connotative syntactic
		Ecological	Metaphorical		

An earcon is based on a *motive* (with the same meaning as is used in music). Rhythm and pitch are the basic memorable components of a motive, and—as in music—a motive exhibits Gestalt characteristics that make it simple, unique, recognizable, and irreducible. The designer then creates related earcons by applying standard musical development techniques—e.g., repetition, inversion, and similar variations—to generate individual sounds that are recognizably distinct and yet noticeably related. The relationships make earcons useful auditory cues to represent hierarchies, networks, and other chunked structures.

Variations on a motive include combination, transformation and inheritance. Earcons, like icons, can be combined to produce compound earcons, and the individual elements that make up an earcon can be multiplied, aligned, shifted in terms of some parameter (e.g., timbre, register) and otherwise varied. Blattner et al. (1989), for example, describes a three-level error tree using family earcons that together present a structured auditory description of computer errors.

Brewster and Crease (1999) demonstrate that sound can be an effective secondary modality to assist with usability of visual menus. In particular, four common errors—item slip, menu slip, slip onto a divider or disabled item, and mis-selection—can be noticed and corrected more quickly in systems that use earcons as feedback on the timing of mouse-pointer movements. Going the reverse direction, speech recognition errors can sometimes be noticed and corrected more quickly with visual feedback. Balentine (1994) describes *callout* menus, *time-cascaded* windows, and *backtime* buttons—all devices for pointing at speech events from the immediate past to maintain dialogue momentum in the face of ambiguities (low confidence) or outright recognition errors (false acceptance or substitution).

Abstract Sounds

Many metaphorical and symbolic sounds, including most earcons, are abstract. A continuously-rising Shepard-Risset glissando, for example, might be associated with moving a scrollbar up. Gaver would call this *metaphorical*, and Keller/Stevens would agree, adding that it is *indirect*. Metaphorical in that "upward movement" of the sound correlates with upward movement of the scroll bar, but there is no other more natural or physical connection between the two. Indirect in that a moving document and a moving sound are not causally-related, but the percept of *rising and falling* is still applicable to both. See Yalla and Walker (2007, 2008) for more on advanced auditory menus and auditory scroll bars.

Spearcons

One of the more exciting innovations in speech output and sonification is the spearcon, designed and tested by Bruce Walker and his students at the Georgia Tech sonification lab. A spearcon starts as a spoken phrase—either a recorded voice or an audio phrase generated by a TTS synthesizer. The audio is then time-compressed without modifying the pitch.

A spearcon can be so compressed that it is no longer comprehensible as a word or phrase; in fact, even to the point where it is no longer recognizable as speech. This makes it an abstract sound that is as flexible as an earcon in terms of its symbolic utility, but also as direct and denotative as an auditory icon.

Walker et al. (2006) compared the spearcon to a fingerprint. This is because, "each unique word or phrase creates a unique sound when compressed that distinguishes it from other spearcons" (Palladino and Walker 2008). This makes spearcons and their referents easy to learn and easy to recognize (Palladino and Walker 2007). One is tempted to hypothesize that the sonic pattern retains the speech attributes of its etymon while acquiring new attributes as an abstract sound-signal. This creates the exciting possibility that a spearcon can *co-exist* in close proximity with speech (as music does), while exhibiting both nomic and syntactic powers.

Blattner et al. (1989)—when discussing icons as the visual counterpart of earcons—observes, "There is one striking difference in the applications of icons and earcons at the present time. Icons are both selectable and informational, whereas earcons are informational only." The greatest HCI potential for spearcons may be in an SNLD dialogue that allows a user to *speak* the (unaltered) word or phrase—thus making the spearcon *selectable* as well as informational. This makes spearcons potentially more useful than either earcons or auditory icons as *tags or labels* offered to the user for speech selection, *contiguous prompts* in faster-than-real-time menus, *ShadowPrompt*® interjections, or multimodal *monikers* (which I will discuss shortly).

Lists

A WIMP list is a series of information chunks—presented in graphical or textual form. Each chunk is called an element. The radio button is an exclusive-element selector. The check box is an inclusive-element selector (Fig. 14.5).

Audio lists are handled in a similar fashion. I have tested several general-purpose list-presentation and element-selection devices—tuned for speech but with native support for other modalities. Features are designed to correlate as faithfully as possible with the radio button/checkbox paradigm while taking advantage of audio metaphors.

- The list is presented as spoken words or spearcons;
- The list can be arbitrarily long;

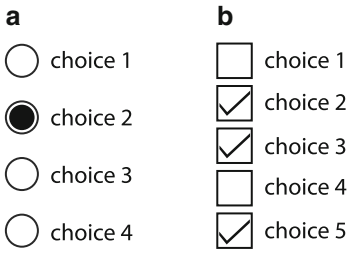

Fig. 14.5 Two common WIMP list-management devices. Radio buttons (**a**) are exclusive, and check boxes (**b**) are inclusive

- Hot zones are represented by time windows (Blattner et al. 1992);
- If the user starts her speech within any portion of a given time window, then the voice is pointing at that element (Balentine 2001);
- The beginning and the end of the list have special turn-taking rules, but there are no turn-taking issues within the list—transition from one time window to the next is seamless;
- The *fastest* way to select any element is to speak it (as soon as you know what to say) and be recognized with high confidence; this includes speaking at the very start of the list (repeat users);
- The *most accurate* way to select any element is to speak it immediately after hearing it and to be recognized with high confidence;
- There are several alternatives for selection that allow for uncertainty or convenience with the tradeoff of lowering speed, accuracy, or both; yes-no questions (Pitt 2003) are especially important, and must be well-crafted and fast;
- Selection alternatives provide a certain degree of *discoverability*;
- Demonstrative OOG (e.g., "that one") provides the most robust selection alternative for novices (Balentine 1999); and,
- Various forms of multimodal selection are also supported.

Audio lists with the above properties—using both spoken and manual interaction—are effective for phonebook/address lists, recorded names, restaurant-selection, train- and airline-scheduling, calendar appointments, and many other list-selection data types. EIG has been focused on telephone-based IVR dialogues, and so most such lists are short (3–20 elements). But internal work and certain client research has shown promising results with long (25–50 elements) and very long (hundreds of elements) lists.

Palladino and Walker (2008) explored 50-element contact lists with visual, auditory, and mixed media. One of the interesting findings involved user learning, which was much more pronounced for non-visual (audio-only) list navigation than with visual media. It is tempting to conclude that *learnability* is a more important feature of an auditory interface than it is in a visual environment—both because an auditory interface is innately slower, and because users bring less experience to the auditory interaction.

This last point about learnability is especially important in an SNLD interface that is *owned* by the user. Most experience with spoken menus is with IVR, which

is *victim automation* that includes little user choice and must assume no prior user experience. An SNLD dialogue that can assume user opportunity to learn—and that affords the designer alternatives for instruction—can play by very different rules as long as user learning is demonstrably fast.

Highly-Directed One-Shot Dialogues

Now that we have speech-and-audio equivalents for all of the major WIMP widgets, we can assemble them into dialogues that give us coherent, linguistically-relevant SNLD behaviors. The primary attributes of such dialogues are:

- Highly directed;
- Very fast;
- Exceptionally robust ("accurate");
- Intrinsically multi-tasking;
- Thoroughly-integrated with WIMP modalities; and,
- Tightly-personalized.

These attributes are best understood with tasks that can be accomplished through short one-shot dialogues. The three examples that we use here are the timer, the clock, and the currency converter. These simple dialogues share certain common characteristics that point toward the advantages of SNLD design. We will call them *gadgets*, as they correspond exactly with the utility gadgets available for Windows and Mac computers.

Soundspace

With these highly-directed multimodal devices in hand, the next question is where to put them. This should not be a problem—after all, sound does not take up any space at all.[4] But the invisibility of speech and sound can be a drawback as well as a feature: things that you can't see don't exist. Methods for calling them into existence must be learned, and if the interface is to be multimodal, then the user must have some mechanism for thinking of sound objects in reference to the GUI (visual) space.

Multimedia computers are equipped with stereo sound already available. The speakers are usually located to the left and the right side of the computer display, and may be thought of as *extending the display's horizontal space*. The line represented

[4] Not exactly true, sound waves must propagate through space to the eardrum before a human user can hear them. So at least the body-sized envelope of sound-passing air (in the case of open-air loudspeakers), or the head-sized envelope implied by headphones or ear buds define the minimum spatial requirements for HCI audio. But because sound perception is so abstract—and because the user cannot see the sounds—the statement is practical.

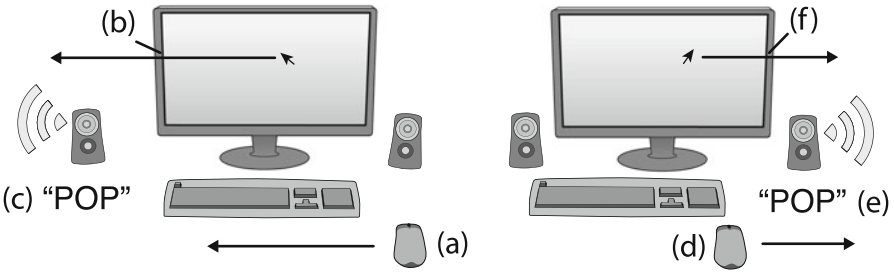

Fig. 14.6 Pointing into soundspace creates the illusion of an abrupt transition between visual and auditory modalities while retaining a constant spatial reference. The user moves the mouse to the left (**a**) or right (**d**) to point (**b**, **f**) into left-hand (**c**) or right-hand (**e**) soundspace

by the edge of the video monitor thus separates visual space from auditory space—or soundspace—allowing the user to visualize sounds as real objects.

Shown in Fig. 14.6, soundspace is virtual. To point into left-hand soundspace, the user moves the mouse (a) to the left, causing the graphical pointer on the video display to move in a corresponding fashion. When the pointer reaches the left-hand side of the display (b), it is suppressed—the user can no longer see it. At the same time, a “pop” sound appears at the left speaker (c). To the user, the pointer has passed across the left-hand display boundary and is now pointing off the screen: she is pointing into soundspace.

When the mouse is moved rapidly to the right again, the left-hand speaker again plays a “pop” sound and the graphical pointer reappears on the left side of the display. The user has crossed the boundary from soundspace back into visual space.

In a symmetrical fashion, the mouse may be moved to the right (d). When the pointer reaches the right-hand edge of the display (f), it pops across the boundary into soundspace, disappearing from view and appearing as sound in the right speaker (e). The user may freely move from the right audio channel, through the separating boundary, across the visual display, and to the left audio channel, popping from sound-to-sight and then -sound again as the mouse pointer traverses the now-enlarged virtual working space.

The Grid

Once in soundspace, the user may browse about spatially, moving the mouse forward and backward to create vertical motion. Soundspace is graduated, with eight tick marks equidistant between top and bottom of the display. The ticks are short MIDI sounds that in sequence form a diatonic major scale. In other words, users do not *see* the grid, they *hear* it.

Fig. 14.7 Audio grid uses musical tones to imply vertical locations within soundspace. The implied grid is shown as eight audio tick-tones in the key of E Major (treble clef not shown). Each note corresponds to a tick on the grid

After the user points into left or right soundspace, moving the mouse “vertically” (which is to say forward and backward) causes a tick-tone to play whenever the imaginary pointer crosses a grid line. Vertical space is implied by pitch. “High” tones are at the top of the screen; “low” tones are at the bottom.

If the user moves the mouse vertically in the right-hand soundspace from the bottom to top and back down, an ascending and descending major scale emerges from the right speaker. The tempo of the scale reflects the speed at which the mouse is moved. Even though the sounds are coming from the same source—the right-channel speaker—the user perceives that an invisible audio pointer is “moving up and down” within the space (Fig. 14.7).

The same two-dimensional illusion applies to the left channel. The user moves the mouse horizontally, popping into left-hand soundspace. Once there, vertical movement within soundspace generates the audio grid with diatonic tick-tones from the left-channel speaker.

The Soundstrip

Moving now from the desktop to a smartphone or tablet, the mouse is replaced with a touchscreen for direct user pointing. The soundspace grid is slightly altered to support this variant. In Fig. 14.8, the phone sports a *soundstrip* (a)—a set of tick marks that spill from the phone casing onto the display—and the corresponding auditory tick-tones (b).

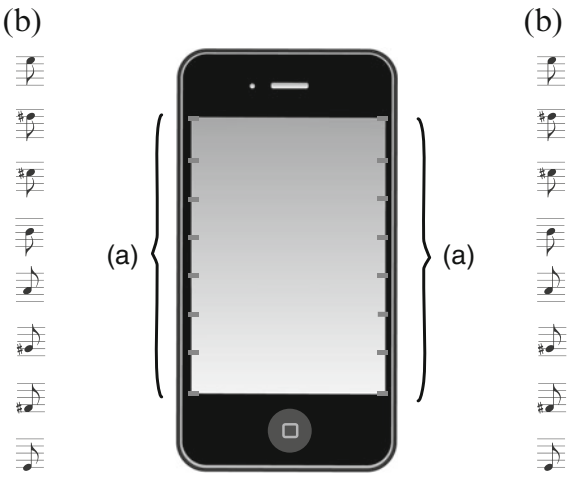

Fig. 14.8 Soundspace on a smartphone. Because there is no pointing device, the grid ticks are indicated visually with a soundstrip (**a**). Just as with the desktop, auditory tick-tones (**b**) imply vertical soundspace

Frame Dimples

In a real embodiment, the smartphone casing itself might include a reference to the soundstrip. For example, concave dimples or convex bumps may be stamped into the physical material.[5] Such a physical reference would be especially useful where sight is limited—either because the user is blind, or because environmental circumstances limit the utility of vision (Fig. 14.9).

Touchpads

Adjacent to each frame dimple, soundstrip ticks are carried directly onto the display with *touchpads* that are very narrow (a few pixels) and tall enough to align well with the dimples. As the user runs a finger up and down the soundstrip, the finger spills onto the edge of the display and activates the touchpads. The physical dimples plus the visual touchpads constitute a multimodal soundstrip that occupies almost no GUI real estate but still allows pointing into soundspace (Fig. 14.10).

Auditory Tick-Tones

There are four ways a user may listen to the soundspace grid, from lowest fidelity to highest:

- Mono lo-fi;
- Open stereo;

[5] Remember that this description is focused on maximum integration of auditory media with other modalities. There are many ways to accomplish such integration, and this method, of course, presents practical product-development obstacles. But it is conceptually easy to understand, and thus makes a good example for the purposes of this article.

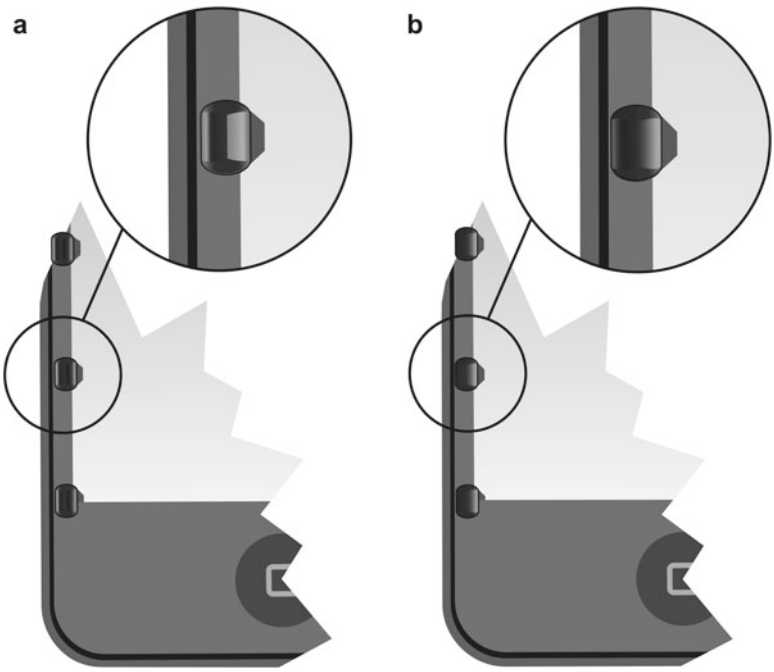

Fig. 14.9 Dimples embossed into the frame may be concave (**a**) or convex (**b**), allowing tactile reinforcement of the grid ticks

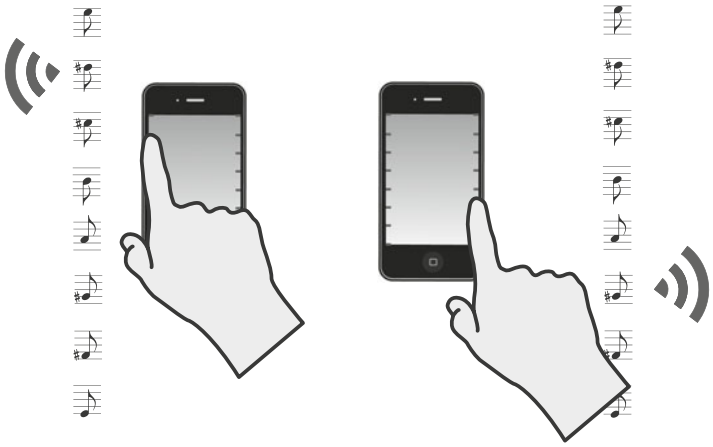

Fig. 14.10 Pointing into soundspace entails touching one of the touchpads on the soundstrip. The leading edge of the touch triggers the corresponding tick-tone

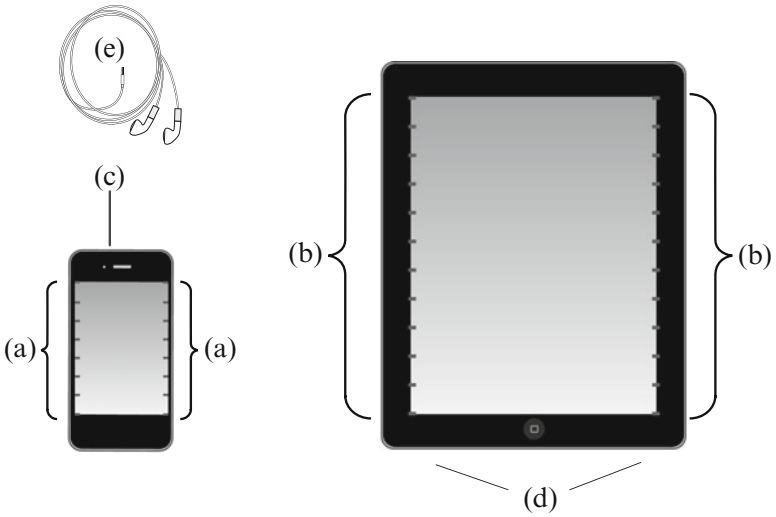

Fig. 14.11 Eight auditory tick-tones create a major scale on smartphones (**a**) 13 ticks a semitone apart create a chromatic scale on the larger tablets (**b**) Playback is through a single mono speaker (**c**) or stereo speakers (**d**) More vivid spatial illusions require earbuds (**e**) or 3D surround audio signal processing (Fig. 14.12)

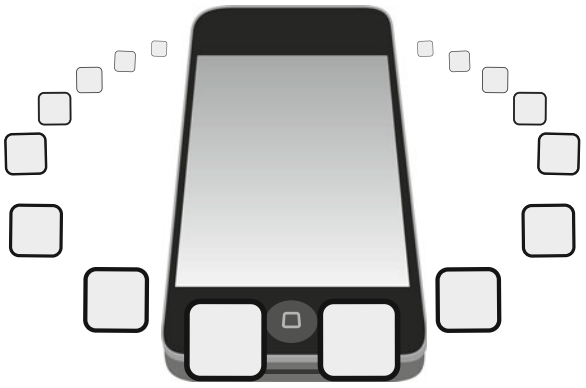

Fig. 14.12 3D Soundspace is made more vivid by spatial-positioning software that places the objects into a 3D surround context

- Closed stereo; and,
- 3D surround.

Mono low-fidelity in Fig. 14.11c plays auditory ticks and other sounds through the single loudspeaker on the device. The user must infer "up and down" and "left and right" through pitch polarity and moniker placement. Open stereo (d) is for those devices that have two speakers. Closed stereo (e) relies on the traditional earbud or headphones—an increasingly common configuration, since these devices are also used for listening to music.

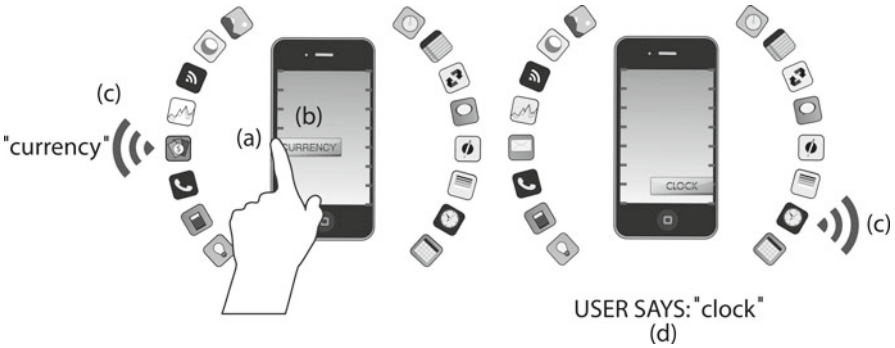

Fig. 14.13 Interactive speech objects reside in soundspace at each grid tick on the soundstrip. The user may point at a grid tick (**a**) to see (**b**) and hear (**c**) the object's moniker

Interactive Speech Objects

The video-audio boundary, represented by graphical-pointer suppression coupled with the pop sound, and the audio grid, represented by the high and low pitches of audio tick-tones, together form the illusion called soundspace. It is an audio equivalent to the video desktop metaphor. Like the desktop, soundspace doesn't do anything. It is merely a place to put things.

The objects deposited into soundspace are interactive speech objects. Each object has as its goal the completion of some extremely bounded task. Such tasks may include telephone answering, voice dialing, message taking, voice annotation, text scanning, and similar human-computer interactions.[6] The objects may therefore be conceived as applets that are uncoupled from and bear no relationship to any foreground application.

One object may be located at each grid tick on the soundstrip. As shown in Fig. 14.13, the user may point at a grid tick (a) to see (b) and hear (c) the object's moniker. After pointing, the object is momentarily open: the user may continue with either a highly-directed or NL dialogue according to preference and skill. After learning, the user launches by speaking (d) the object's name (represented by the moniker), making interaction fully hands-free.

Monikers

A moniker is a multimodal label and attachment-point/handle for the multimodal gadgets that reside in soundspace. Standard GUI gadgets (which exploit the three dimensions of space) are represented by icons, and we as users already know how to launch, minimize, resize, drag, and dismiss them. Multimodal gadgets—although they have the same GUI features—also support modalities associated with the dimension of time (primarily sound). Temporal manipulation techniques are not yet possessed by a large population of users. The moniker provides a standard tool for learning them.

[6] These directed dialogues are well-known in the speech industry and not labored here.

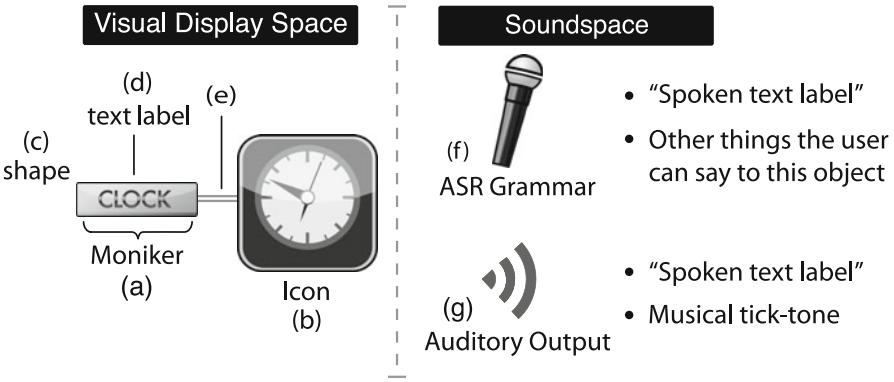

Fig. 14.14 Moniker (**a**) attached to its icon (**b**). Visible components include shape (**c**), text (**d**), and attachment point (**e**). Invisible components include an ASR grammar (**f**) and auditory output (**g**)

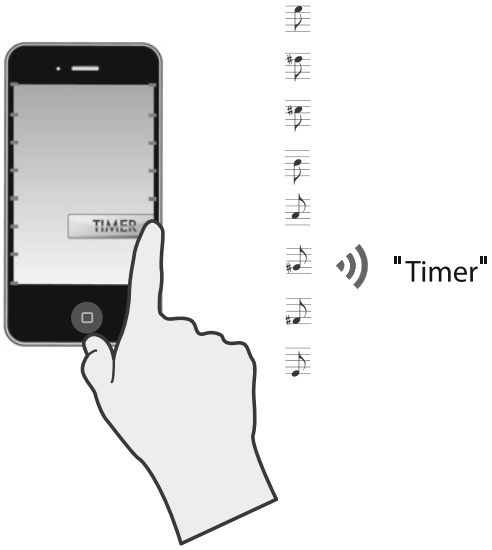

Fig. 14.15 Initial press of the touchpad triggers the tick-tone. Continuing to hold the touchpad then triggers—a fraction of a second later—the spearcon that speaks the label. Concurrent with playback, the visual moniker appears on the display

Shown in Fig. 14.14, the GUI components of a moniker (a-d) are familiar. With them we can *see* and *touch* the moniker, as well as *grab* it to move the gadget. The auditory components reside in soundspace. With them, we can *hear* and *speak* to the moniker to perform similar operations. The moniker speaks its label to the user via an auditory spearcon. Visual objects such as icons can be seen, so the moniker attaches to icons in a traditionally visual way—the shape with its text connects via a line to an attachment point on the icon (e). Auditory objects cannot be seen, so there is nothing visual to attach the moniker to. Therefore, we use the soundstrip ticks as attachment points (Fig. 14.15).

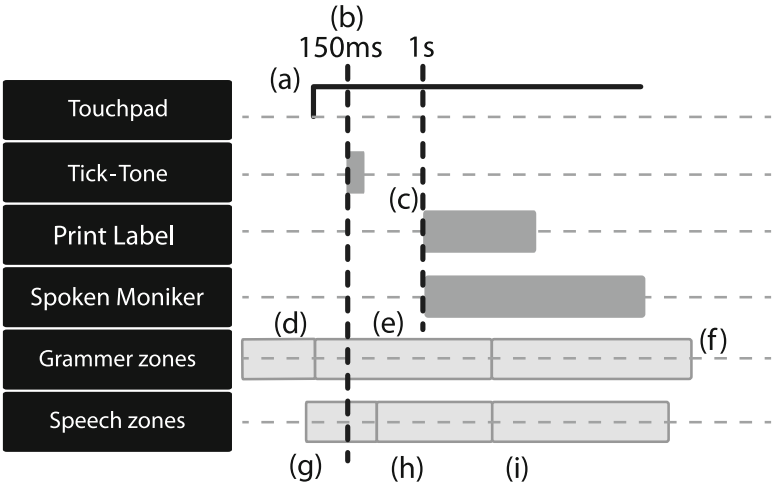

Fig. 14.16 Touchpad timing. The leading edge of a touch (**a**) triggers the tick-tone, followed moments later by the moniker and its spoken-label spearcon (**c**)

When a gadget is on the display, the moniker is attached to its icon, and we see both superimposed over their background. When a gadget resides in soundspace, it is completely invisible except under specific circumstances. Under those circumstances, the moniker makes itself visible and we can manipulate the gadget through its moniker. Think of the moniker as a "porthole" through which we can see, hear, touch, and talk to the gadget.

The user touches the touchpad (a). The short delay (b) is designed to ignore spurious triggers. Once it is clear that the touch is intended, the short tick-tone plays. If the user releases the touch quickly, then the tone is all that plays. But the tone always finishes playing. This allows the user to run a finger up and down the strip to hear the grid. The grammar is initially in its quiescent gatekeeper state (d), awaiting launch words for any of the active gadgets. Once the touch modality indicates a possible user interest, the grammar reflects the changed expectation (e) that this gadget is the one of interest. If the user holds the touchpad long enough to hear the moniker (c), then the ASR attention is focused more tightly on speech relevant to the currently-selected gadget (f). In addition, the onset of speech (g, h, i) modulates expectations and post-recognition analysis[7] by monitoring short-term user memory to infer user intention (Fig. 14.16).

Populating Soundspace

To populate soundspace, the user first downloads a gadget, for example the clock shown in Fig. 14.17a. Such applets will likely reside on a third-party remote website.

[7] Post-recognition analysis includes *n*-best list traversal, interpretation of confidence values, and other executive decision-making.

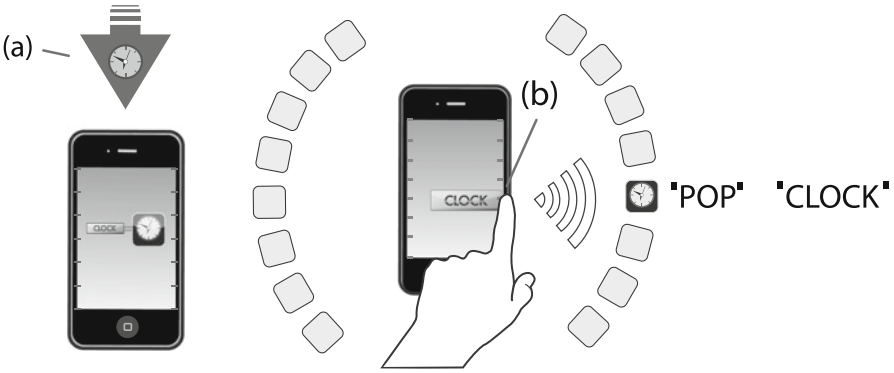

Fig. 14.17 The user populates soundspace by downloading a gadget (**a**) and dragging it to some unused location in soundspace (**b**)

The user then drags the gadget to some unused location in soundspace (b). When the finger arrives at the soundstrip touchpad, the corresponding tick-tone appears as auditory feedback, and the icon disappears—leaving only the moniker attached to its soundstrip tick. On release of the finger, the gadget plays a "pop" followed by its moniker spearcon. Moniker and icon then disappear.

Except for playing its moniker when requested, a speech object does not generally make its presence known. Specifically, there is no iconographic representation or graphical window occupying the video work space. Indeed, a major feature of this scheme is the reclamation of valuable screen space without an attendant loss of productivity—time normally consumed when the user must minimize, move, or shuffle video objects about on the windowing surface. Tasks that occur infrequently, but are useful only if ready-to-hand (Winograd and Flores 1986) thus become major candidates for shifting into soundspace.

Interacting with Objects

The user interacts with objects in soundspace primarily by speaking to them. Each object has a launch- or start-word represented by its moniker. The user may say, "Timer," for example, to activate the timer object. Upon activation, the object moves the user through a sequential dialogue aimed at accomplishing the user's goal (see the Timer audio examples).

In a similar fashion, the moniker, "Clock" launches a dialogue with the clock gadget. In addition to announcing time and date, the object may offer spoken information regarding impending appointments or reminders. Other launch words trigger dialogues with speech objects that read e-mail, play v-mail, report on internal tasks, capture spoken voice reminders, schedule events, or activate proof-reading voice scanners.

The fact that they are grammatically trivial, simple, and bounded, entailing extremely small vocabularies at a given state, speech gadgets are useful primarily

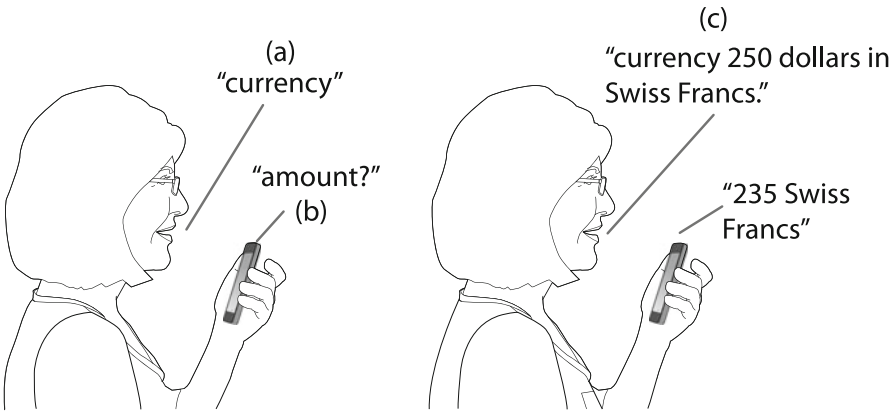

Fig. 14.18 Saying the moniker launches a speech object (**a**). Unknowledgeable users are then prompted for the next chunk of information (**b**) quickly and minimally. Knowledgeable users may speak multi-token NL sentences (**c**) to suppress the highly-directed prompting (**b**), but such utterances are not required nor expected

for mundane and un-taxing background tasks. Complexity, as it arises over time, appears in the sheer number of such specialized objects, and not in the difficulty of any one. For this reason, user learning is spread over time, and perceived complexity remains low. When interacting with a given object, the functions represented by the others are not within the user's problem domain and do not occupy user attention (Fig. 14.18).

Conclusions: The Integrated 4D WIMP

The overall goal of the model described in this article is a complete integration of the dimension of time into the traditional WIMP paradigm—that is, a true 4D standardized interface. By considering 4D WIMP as an achievable goal, we solve many short-term problems that are currently intractable in the traditional NL speech space.

There are several HCI benefits to the integration model just described:

- Instant availability;
- Easy start with directed learning;
- Transfer of learning;
- Abandonment without penalty;
- Structure and order in the absence of visual cues;
- Multiple reinforcing referents;
- Minimum impact on spatial modalities; and,
- User choice of modalities and media.

Instant availability is the first and most important benefit. All objects in soundspace are one gesture away, and that gesture may be manual or spoken. There is no competition between GUI objects on the display space and the sound objects in soundspace. *Easy start*, in turn, allows the same "user-first" model of interaction as that supported by fully NL designs.

Transfer of learning derives from the widget model that emphasizes reuse. Every dialogue that requires that the user record some spoken information—for either short- or long-term purposes—uses the same recording device. Behaviors are the same whether speaking phonebook names, grocery lists, map directions, ideas for a story, thoughts during a drive, or a guest list for an upcoming party. Similarly, any such voice recording—or sets of recordings—may be reviewed using the sonic window. Users learn meta-behaviors once and forever, and not on a topic-by-topic basis.

Abandonment without penalty replaces laborious error-recovery routines so common in most directed dialogues. In one clever solution to this problem (Rennyson and Bouzid 2012), the user simply *shakes the phone* to abandon the dialogue—the equivalent of erasing an Etch-a-Sketch®. *Structure in the absence of visual cues* assists: when you shake the phone to erase a dialogue-gone-wrong, the gadget remains active in soundspace—it's simply a matter of starting it back over again. This is a much lower penalty than fully closing a GUI gadget.

Multiple reinforcing referents derives from the multileveled nature of audio. The sonic window grid ticks, for example, have an iconic impact as simulated "clock ticks" while concurrently providing syntactic reinforcement through the harmonic progression implied by the music. Similarly, the "up and down" polarity of the soundstrip ticks reinforce their physical arrangement on the display casing. Both, in turn, reinforce the 3D placement of gadgets in soundspace.

Minimum impact is a marketing issue, because existing WIMP operating systems are in place and well-evolved. The SNLD model calls for a deep integration of temporal media including speech, and will likely introduce disorder into this well-established ecosystem. Care in interdisciplinary design, plus attention to the political dynamics of intrusion by speech and sound experts into the visual and manual domain are essential for speedy development.

Regarding *user choice*, if immediate circumstances call for the auditory modality, then the user will choose that modality and the design must serve it without bias. If the auditory modality is not essential but preferred, then user personality and circumstances will determine that choice, and the user interface need not second-guess. If the auditory modality is inappropriate, then the user will certainly choose one of the eye-hand coordination modalities available—without requiring or expecting the user interface to assist in that decision. In other words, a broadly-generalized 4D WIMP expands the options for user choice without expanding the intelligence requirements (and errorfulness) of the operating system. Media are available in parallel, and the system turns the modality decision over to the user.

Closing

A few final comments regarding urgency: we in the NL speech community must understand that advanced auditory menus, sonic windows, and other super-natural HCI are advancing just as fast as NL research. Moreover, alternative non-speech modalities, e.g., collapsible and folding touch displays, projected virtual keyboards, gesture-in-empty-space, and similar advanced HCI devices are competing with speech for user attention.

Finally, screen space that is required for integration of SNLD into GUI interfaces is being rapidly claimed by other modalities for their own uses, continuing to push speech into the “standalone modality” straitjacket. Now is the time for the WIMP, sonification, ASR, NL, and interactive communities to make a concerted interdisciplinary effort toward integration.

By describing a complement to standalone NL that fits today’s WIMP paradigm, I have shown that there is a way to apply super-natural speech design principles to a multimodal interface. In this model, super-natural gadgets fill the need for repetitive and common tasks, while standalone NL solutions are tailor-made to the unique, one-of-a-kind, intelligent tasks for which they are best-suited. The former live in soundspace, and the latter hover “above” and “around” the device with no spatial restrictions.

Acknowledgments Illustrations by Alexander T. Klein. Spearcons and Spindex are trademarks of Georgia Tech University. Etch-a-Sketch is a registered trademark of The Ohio Art Company. ShadowPrompt is a registered trademark of Enterprise Integration Group.

References

Arons B (1994) Interactively skimming recorded speech. Dissertation, MIT Press, Boston

Balentine B (1994) A multimedia interface: speech, sound, sight, and touch. In: AVIOS ’94 Proceedings, San Jose, Sept 1994

Balentine B (1999) Re-engineering the speech menu. In: Gardner-Bonneau D (ed) Human factors and voice interactive systems. Kluwer Academic, Boston, pp 205–235

Balentine B (2007a) It’s better to be a good machine than a bad person. ICMI Press, Annapolis

Balentine,B (2007b) Online articles regarding menu guidelines for HCI. Click on “Lists and User Memory”. http://www.beagoodmachine.com/extras/cuttingroomfloor.php

Balentine B, Morgan DP (2001) How to build a speech recognition application. EIG Press, San Ramon

Balentine B, Melaragno R, Stringham R (2000) Speech recognition 1999 R&D program final report: user interface design recommendations, EIG Press, San Ramon

Blattner MM, Greenberg RM (1992a) In: Edwards ADN, Holland S (eds) Multimedia interface design in education, Springer, Berlin

Blattner MM, Sumikawa DA, Greenberg RM (1989) Earcons and icons: their structure and common design principles. Hum Comput Interact 4:11–44

Brewster SA, Crease MG (1999) Correcting menu usability problems with sound. Behav Info Technol 18(3):165–177

Buxton W, Baecker R, Arnott J (1985) A holistic approach to user interface design. Unpublished manuscript
Christian B (2011) The most human human: what artificial intelligence teaches us about being alive. Anchor Books, New York
Commarford PM, Lewis JR, Smither JA-A, Gentzler MD (2008) A comparison of broad versus deep auditory menu structures. Hum Factors 50(1):77–89
Gaver WW (1986) Auditory icons: using sound in computer interfaces. Hum Comput Interact 4:167–177
Keller P, Stevens C (2004) Meaning from environmental sounds: types of signal-referent relations and their effect on recognizing auditory icons. J Exp Psychol Appl 10(1):3–12
Koelsch S (2012) Neural correlates of processing musical semantics. In: First international workshop on segregation and integration in music and language, Tübingen, Feb 2012
Kramer G (1994) An introduction to auditory display. In: Kramer G (ed) Auditory display: sonification, audification, and auditory interfaces. Addison Wesley, Reading, pp 1–78
Kramer G, Walker B, Bonebright T, Cook P, Flowers JH, Miner N, Neuhoff J (2010) Sonification report: status of the field and research agenda. Lincoln: http://digitalcommons.unl.edu/psychfacpub/444
Lewis JR (2011) Practical speech user interface design. CRC Press, Boca Raton
Martin P, Crabbe F, Adams S, Baatz E, Yankelovich N (1996) Speech acts: a spoken language framework. Computer 29(7) IEEE Computer, pp 33–40
Meyer LB (1956) Emotion and meaning in music. The University of Chicago Press, Chicago
Miller G (1956) The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol Sci 63:81–97
Muller M, Farrell R, Cebulka K, Smith J (1992) In: Blattner M, Dannenberg R (ed) Multimedia Interface Design. ACM Press, Michigan, p 30
Palladino DK, Walker BN (2007) Learning rates for auditory menus enhanced with spearcons versus earcons. In: International conference on auditory display, Montreal
Palladino DK, Walker BN (2008) Efficiency of spearcon-enhanced navigation of one dimensional electronic menus. In: International conference on auditory display, Paris
Pitt I, Edwards A (2003) Design of speech-based devices. Springer, London
Rennyson D, Bouzid A (2012) Personal conversation. Virginia, Vienna
Serafin S, Franinvić K, Hermann T, Lemaitre G, Rinott M, Rocchesso D (2011) Sonic interaction design. In: Hermann T, Hunt A, Neuhoff JG (eds) The sonification handbook. Logos-Verlag, Berlin, pp 87–110
Suhm B, Freeman B, Getty D (2001) Curing the menu blues in touch-tone voice interfaces. In: Proceedings of CHI 2001 , ACM, The Hague, pp 131–132
Walker BN, Nees MA (2011) Theory of sonification. In: Hermann T, Hunt A, Neuhoff JG (eds) The sonification handbook. Logos-Verlag, Berlin, pp 9–39
Walker BN, Nance A, Lindsay J (2006) Spearcons: speech-based earcons improve navigation performance in auditory menus. In: Proceedings of the international conference on auditory display (ICAD 2006), London(June 20–24), pp 63–68
Winograd T, Flores F (1986) Understanding computers and cognition. Addison-Wesley, Menlo Park
Yalla P, Walker BN (2007) Advanced auditory menus. (No. GIT-GVU-07-12.): Georgia Institute of Technology GVU Center
Yalla P, Walker BN (2008) Advanced auditory menus: design and evaluation of auditory scroll bars. In ASSETS'08, ACM Press, Halifax

Editors' Biographies

Amy Neustein, Ph.D., is editor-in-chief of the *International Journal of Speech Technology* and Series Editor of the *Springer Briefs in Speech Technology*. She has edited two prior Springer books: *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics* and *Forensic Speaker Recognition: Law Enforcement and Counter-Terrorism*. She is a member of the visiting faculty at the National Judicial College since 1985, and a member of MIR (Machine-Intelligence Research) Labs since 2010. She is the recipient of several distinguished awards: pro Humanitate Literary Award; Information Technology: New Generations (Medical Informatics) Award; the Los Angeles County Supervisor Humanitarian Award; and the Woman of Valor: Lifetime Achievement Award. She is the CEO and Founder of Linguistic Technology Systems, located in Fort Lee, New Jersey.

Judith Markowitz, Ph.D., has been a leading analyst and thought-leader in the speech-processing industry for over 25 years. She is an invited expert to the World Wide Web Consortium (W3C)'s Voice Browser Working Group. She has served on two American National Standards Institute (ANSI) committees and was made lead editor of INCITS 456, an ANSI standard for voice biometrics. In 2003, *Speech Technology Magazine* named her one of the top ten leaders in speech; in 2006 she was elevated to IEEE Senior Member status. In addition to her work in speech processing, she is national president of the Lambda Literary Foundation. She is the president of J. Markowitz, Consultants, located in Chicago, Illinois.

A. Neustein and J.A. Markowitz (eds.), *Mobile Speech and Advanced Natural Language Solutions*, DOI 10.1007/978-1-4614-6018-3,
© Springer Science+Business Media New York 2013