1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999

1989-1999 1989-1999

LABORATORY *for*
COMPUTER
SCIENCE

July 1999

SPOKEN
LANGUAGE
SYSTEMS

1989-1999 1989-1999

1989-1999 1989-1999

1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999

MIT
Massachusetts
Institute
of Technology

©1999 Massachusetts Institute of Technology

For information or copies of this report, please contact:

Victoria L. Palay
MIT Laboratory for Computer Science
545 Technology Square, NE43-601
Cambridge, MA 02139 USA
palay@mit.edu

Please visit the Spoken Language Systems Group on the
World Wide Web at http://www.sls.lcs.mit.edu

10th Anniversary Issue

SUMMARY *of* RESEARCH

covering

July 1, 1998 - June 30, 1999

Table of Contents

1 Summary of Research

Research, Technical, Administrative and Support Staff .... vii-ix
Graduate Students .... x-xii
Visitors, Undergraduate Students, Transitions .... xiii-xiv
Research Sponsorship xv

2 Research Highlights

Research Highlights, 1998-1999
Victor Zue 3

3 Research Projects

GALAXY-II
Stephanie Seneff, Raymond Lau, Christine Pao and Nikko Ström 9
Multilingual JUPITER
Stephanie Seneff, Joseph Polifroni,Yasuhiro Minami, and Yi-Chung Lin 17
PEGASUS
Stephanie Seneff and Joseph Polifroni 19
Development of the VOYAGER System
T.J. Hazen 20
The MERCURY System for Flight Browsing and Pricing
Stephanie Seneff, Raymond Lau, James Glass and Joseph Polifroni 23
Finite-State Transducers
Lee Hetherington 29
ANGIE
Stephanie Seneff 31

4 Thesis Research

Heterogeneous Lexical Units for Automatic Speech Recognition
Issam Bazzi 35
Towards Multi-Domain Speech Understanding Using a Two-Stage Recognizer
Grace Chung 37
Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition
Andrew Halberstadt 40
Word and Phone Level Acoustic Confidence Scoring
Simo Kamppari 42
SeMole: A Robust Framework for Gathering Information from the World Wide Web

4

Hyung-Jin Kim 45

Thesis Research (continued)

Analysis and Modeling of Non-Native Speech for Automatic Speech Recognition
Karen Livescu 47
Toward a General Framework for Robust Understanding
Kristine Ma 50
PANDA: Telephone-Based Chinese Conversational System in Weather Domain
Xiaolong Mou 53
Subword-based Approaches to Spoken Document Retrieval
Kenney Ng 56
Toward the Automatic Transcription of General Audio Data
Michelle Spina 59
Mandarin Tone Sandhi and Intonation for Improved Tone Recognition
Chao Wang 62

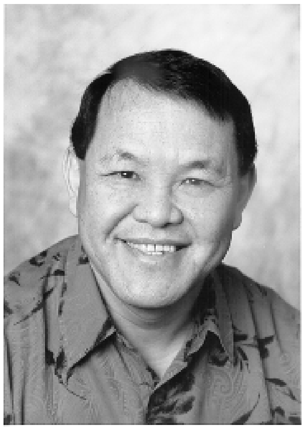
5

Natural-Sounding Concatenative Speech Synthesis of Multi-Sonorant Sequences
Jon Yi 65

Theses, Publications, Presentations and Seminars

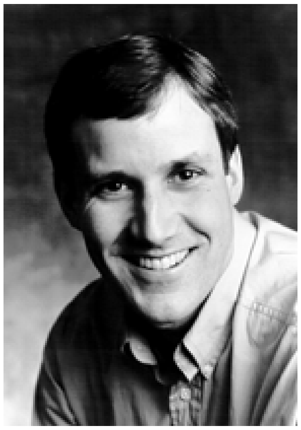
Ph.D. and Masters Theses 69
Publications 70
Presentations 71
SLS Seminar Series 73

Research Staff



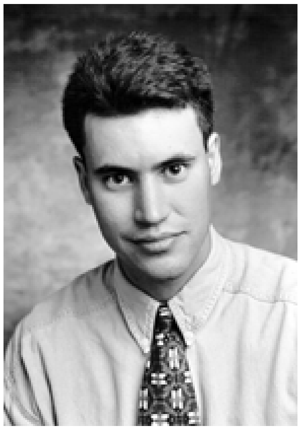
VICTOR ZUE

Victor Zue is a Senior Research Scientist, the head of the Spoken Language Systems Group, and an Associate Director of the Laboratory for Computer Science. His main research interest is in the development of spoken language interfaces to facilitate natural human/computer interactions, and he has taught many courses and lectured extensively on this subject. He is best known for his acoustic phonetic knowledge, and for leading his research group in developing conversational systems that allow users to access information using spoken dialogue. In 1994, Zue was elected Distinguished Lecturer by the IEEE Signal Processing Society. In 1999, he received the "Sustained Excellence" Award from DARPA-DoD. Zue is a Fellow of the Acoustical Society of America. He received his Sc.D. in Electrical Engineering from MIT in 1976.



JAMES GLASS

James Glass is a Principal Research Scientist and Associate Head of the SLS group. He received his Ph.D. in Electrical Engineering and Computer Science from MIT in 1988. His research interests include acoustic-phonetic modeling, speech recognition and understanding in the context of conversational systems, and corpus-based speech synthesis. In addition to publishing extensively in these areas, he has supervised S.M. and Ph.D. students, and co-taught courses in spectrogram reading and speech recognition. He served as a member of the IEEE Acoustics, Speech, and Signal Processing, Speech Technical Committee from 1992-1995. Since 1997, he has served as an associate editor for the IEEE Transactions on Speech and Audio Processing.



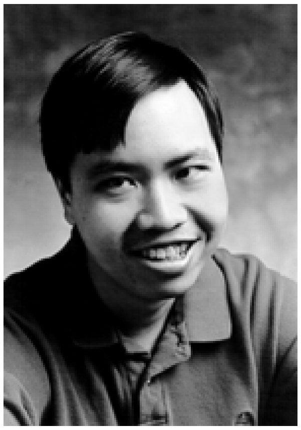
T.J. HAZEN

Timothy James (T. J.) Hazen arrived at MIT in1987 where he received his S.B. degree in 1991, S.M. degree in 1993 and PhD in 1998,all in Electrical Engineering. T.J. joined the SLS group as an undergraduate in 1991 and has been with the group ever since. He is currently working as a research scientist in the group. His primary research interests include acoustic modeling, speaker adaptation, automatic language identification, and phonological modeling.

LEE HETHERINGTON

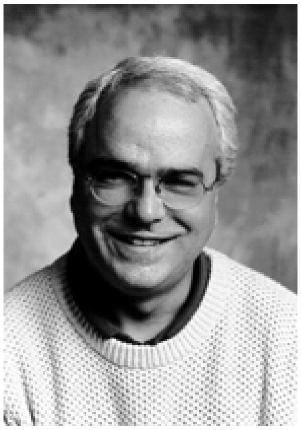
Lee Hetherington received his S.B., S.M., and Ph.D. degrees from MIT's Department of Electrical Engineering and Computer Science. He completed his doctoral thesis, "A Characterization of the the Problem of New, Out-of-Vocabulary Words in Continuous-Speech Recognition and Understanding," and joined the SLS group in October 1994. His research interests include many aspects of speech recognition, including search techniques, acoustic measurement discovery, and recently the use of weighted finite-state transduction for context-dependent phonetic models, phonological rules, lexicons, and language models in an integrated search.

Research Staff



RAYMOND LAU

Raymond Lau received the B.S. in Computer Science and Engineering, the M.S. degree in Electrical Engineering and Computer Science, and the Ph.D. degree in Computer Science, all from the Massachusetts Institute of Technology in 1993, 1994, and 1998, respectively. He was a National Science Foundation fellow and is a member of Eta Kappa Nu. His current research interests are in the are area of speech recognition and spoken language systems with a particular focus on subword modelling, search strategies and language modelling.



JOSEPH POLIFRONI

Joseph Polifroni's interests include language generation, human-computer interaction, and multilingual systems. He has worked on the back-end components of many of the SLS systems, including GALAXY and DINEX in addition to his work on GENESIS, the natural language generation system that is part of the overall GALAXY architecture. He has also contributed to the Spanish and Mandarin Chinese systems. Before joining SLS, Joe worked in the Speech Group at Carnegie Mellon University and was also a consultant for Carnegie Group Inc. in Pittsburgh. In addition, Joe spent two years living in China, teaching English at Shandong University in Jinan.

STEPHANIE SENEFF

Stephanie Seneff has a B.S. degree in Biophysics and M.S., E.E., and Ph.D. degrees in Electrical Engineering and Computer Science from MIT. Her research interests span a wide spectrum of topics related to conversational systems, including phonological modelling, auditory modelling, computer speech recognition, statistical language modelling, natural language understanding and generation, discourse and dialogue modelling, and prosodic analysis. She has published numerous papers in these areas, and she is currently supervising several students at both master's and doctoral levels.

NIKKO STRÖM

Nikko Ström received the Master of Science, (Engineering Physics) degree in1991, and the Ph.D. degree in Electrical Engineering (Department of Speech, Music,and Hearing) in 1997 at the Royal Institute of Technology (KTH), Stockholm, Sweden. He joined SLS in May 1998 as a Postdoctoral Associate, and was appointed Research Scientist in January 1999. His main areas of interest are human/machine dialogue, lexical search in automatic speech recognition, and acoustic/phonetic modeling. At KTH, he developed a continuous automatic speech recognition system that is still in use in several dialogue systems. He is also the author of the NICO tool kit for Artificial Neural Networks. The tool kit is publicly available, and has been downloaded from more than 1,000 different sites in 55 countries (September 1998).

Technical Staff

CHRISTINE PAO

Christine Pao has been a member of the technical research staff since 1992. She is primarily involved in the development and maintenance of the GALAXY system. Her research interests are in discourse and dialog, systems integration with a focus on multilingual systems and language learning, and open microphone issues such as rejection and channel normalization. Christine has a bachelor's degree in Physics from MIT.

Administrative & Support Staff

VICTORIA PALAY

Victoria Palay has been a member of the Spoken Language Systems group since 1988. As SLS program administrator, she manages personnel, fiscal, publication and contractual matters as well as space and other group resources. In addition, she supports Victor Zue's duties as LCS Associate Director by coordinating equipment donations made to the Laboratory. Victoria has a B.A. in Government and French Studies from Smith College.

SALLY LEE

Sally Lee joined the Spoken Language Systems group as senior secretary in 1993. She received a B.A. in Studio Art/ Art History from Colby College in 1984. She also studied at the Art Institute of Boston and the New York Studio School. In addition to her secretarial duties, Sally has made many of the animated and still icons for SLS programs including GALAXY and JUPITER. She also is responsible for transcribing sentences that are recorded from people calling into the JUPITER system.

Graduate Students

ISSAM BAZZI

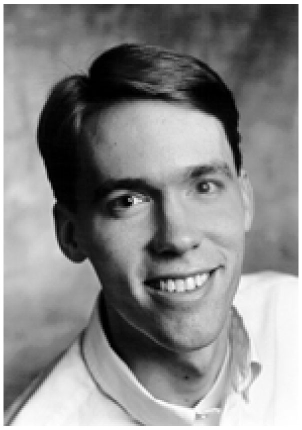
Issam Bazzi's research interest is in the area of subword and language modeling for automatic speech recognition. He received his B.E. in computer and communication engineering from the American University of Beirut, Beirut, Lebanon in 1993 and his S.M. from MIT in 1997. Between 1993 and 1995, He did research on networked multimedia systems at the Center for Educational Computing Initiatives at MIT. Before joining SLS in 1998, Issam was a member of the Speech and Language Department at BBN Technologies, GTE Internetworking where he did research on language-independent character recognition.

Advisor: James Glass

GRACE GHUNG

Grace Chung graduated in Electrical Engineering and Mathematics from the University of New South Wales, Sydney, Australia. She earned a Fulbright scholarship to attend MIT and completed her master's degree in June 1997. Her interests are in acoustic modelling and prosodic modelling for speech recognition.

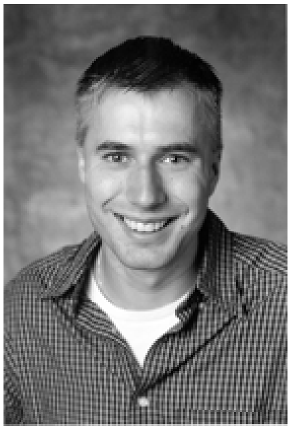
Advisor: Stephanie Seneff



ANDREW HALBERSTADT

Andrew Halberstadt received the B.S. and M.S. degrees in Electrical Engineering from the University of Rochester in 1992 and 1993, respectively. In addition, he received the Bachelor of Music degree in 1991 from the Eastman School of Music in Rochester, NY. He was the recipient of a Sproul fellowship at the University of Rochester, and is a member of the engineering honor society Tau Beta Pi. His research interests include time-frequency representations, phonetic classification and recognition, speech and audio processing, and pattern recognition. Drew completed his Ph.D. thesis in November 1998.

Advisor: James Glass



SIMO KAMPPARI

Simo Kamppari received the S.B. degree in EECS from MIT in 1998 and expects to receive the M.Eng in the fall of 1999. His thesis involves work in word level acoustic confidence scoring and user feedback development in a spontaneous speech understanding system. While a member of SLS, he has studied cellular phone channel effects on speech recognition, developed acoustic confidence scoring methods, and analyzed methods for utilizing confidence scores for improved user feedback.

Advisor: T.J. Hazen

HYUNG-JIN KIM

Hyung-Jin Kim spent his undergraduate years at MIT and completed a Master's of Engineering degree at the SLS group in September 1998. His thesis describes a system called SeMole which is a framework for robustly accessing information on HTML pages. His research interests include Java, XML, and other web technologies.

Advisor: Lee Hetherington

KAREN LIVESCU

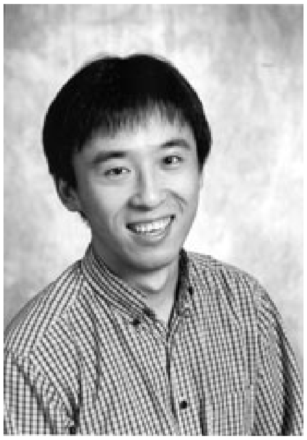
Karen Livescu received her B.A. in Physics at Princeton University in 1996. She spent the following year at the Technion in Haifa, Israel, as a visiting student in the Electrical Engineering department. Karen started graduate study in the SLS group in September 1997. She is a National Science Foundation fellow and expects to complete her Master's thesis in September 1999.

Advisor: James Glass

KRISTINE MA

Kristine Ma joined the SLS Group in September 1998 as a Ph.D. candidate.

Advisor: Victor W. Zue



XIAOLONG MOU

Xiaolong Mou received his bachelor's degrees in Computer Science and Enterprises Management from Tsinghua University, Beijing, China in 1996. He completed his Master's thesis on continuous speech recognition systems and received his master's degree in Computer Science from Tsinghua University in 1998. He is currently pursuing a Ph.D. in the SLS group. Xiaolong's research interest includes conversational speech systems, automatic speech recognition and speech understanding.

Advisor: Victor W. Zue

Graduate Students

KENNEY NG

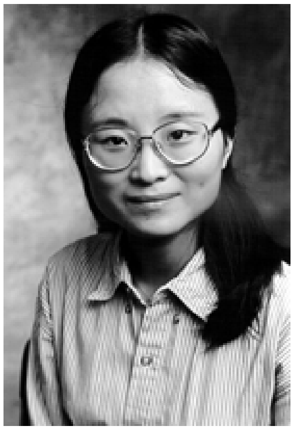
Kenney Ng's current research interest is in the area of information retrieval of spoken documents, which is the task of identifying those speech messages stored in a large collection that are relevant to a query provided by a user. Prior to his return to MIT in 1995, Kenney was a member of the Speech and Language Department at BBN Systems and Technologies where he did research on large vocabulary recognition of conversational speech, word spotting, topic spotting, probabilistic segmental speech models, and noise compensation. He received his B.S. and M.S. degrees in EECS from MIT in 1990.

Advisor: Victor Zue

MICHELLE SPINA

Michelle Spina received the B.S. in Electrical Engineering from the Rochester Institute of Technology in 1991, and the S.M. in Electrical Engineering from MIT in 1994. She is currently pursuing a Ph.D. degree in the SLS group. Michelle's research interests include automatic indexing of audio content, speech recognition and understanding, and biomedical issues of speech processing as they relate to automatic speech recognition. Her current research involves general sound understanding and orthographic analysis of general audio data. Michelle was a 1995 Intel Foundation Graduate Fellow, and is a member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi.

Advisor: Victor Zue



CHAO WANG

Chao Wang received her bachelor's degree in Biomedical Engineering, with a minor in Computer Science from Tsinghua University, Beijing, China in 1994. She started her graduate study in MIT in September 1995 and joined the SLS group in April 1996. Chao's master's degree, completed in June 1997, worked on porting the GALAXY system to Mandarin Chinese.

Advisor: Stephanie Seneff

JON YI

Jon Yi received the S.B. and the M.Eng. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 1997 and 1998, respectively. He also graduated in 1997 with a minor in Music. At SLS he has worked on developing a Mandarin Chinese concatenative speech synthesizer and a UNICODE/Java World Wide Web interface for the GALAXY system. His research interests include speech synthesis, communications systems, and multilingual speech understanding systems.

Advisor: James Glass

Visitors

YASUHIRO MINAMI

Yasuhiro Minami received the M. Eng. Degree in Electrical Engineering and the Ph.D. in Electrical Engineering from the Keio University, in 1988 and 1991, respectively. He joined NTT in 1991. He had worked in robust speech recognition. He joined SLS in 1999 as a visiting researcher. He is interested in modeling for robust speech recognition and building conversational systems. He has been working on porting JUPITER to Japanese.

YI-CHUNG LIN

Yi-Chung Lin received his Ph.D. in Electrical Engineering from the Electrical Engineering Institute of National Tsing Hua University, Taiwan, in July 1995. In his Ph.D. thesis he investigated the use of "A Level Synchronous Approach to Ill-formed Sentence Parsing and Error Recovery". He joined SLS in March 1999 as a Visiting Scientist. He is interested in building Chinese conversational systems for real users. His main focus has been in the area of language understanding, language generation and dialogue management. He has been working on Chinese JUPITER, the weather information system using spoken Mandarin Chinese.

Undergraduate Students

Chian Chuu
Michael Freedman
Jonathan Lee
Mark Knobel
Vivian Ma
Eric Sandness
Rafael Schloming
Archit Shah
Samuel Wong
James Wood
Minnan Xu

Transitions

Andrew Halberstadt, PhD., 11/98
Philipp Schmid, 11/98 joined Microsoft Corp.
Nikko Ström, promoted to Research Staff, 1/99
Ed Hurley, departed 1/99.
Hyung-Jin Kim, M.Eng., 9/98.
Joseph Polifroni, promoted to Research Staff, 7/98

Research Sponsorship

Defense Advanced Research Projects Agency[1]
BellSouth Intelliventures
Industrial Technology Research Institute
National Science Foundation[2]
Nippon Telegraph & Telephone

In addition, discretionary funds for research are provided by ATR Interpreting Telecommunications Research Laboratories, Hughes Research Laboratories, the IBM University Partnership Program, Intel Corporation and Science Applications International Corporation.

---

[1] Contract Nos. N66001-96-C-8526 and N66001-99-1-8904, from the Information Technology Office, monitored by the Naval Command Control, and Ocean Surveillance Center and contract no. DAAN02-98-K0003, monitored through US Army Soldier Systems Command.

[2] NSF grant no. IRI-9618731.

RESEARCH

HIGHLIGHTS

Research Highlights

Victor Zue

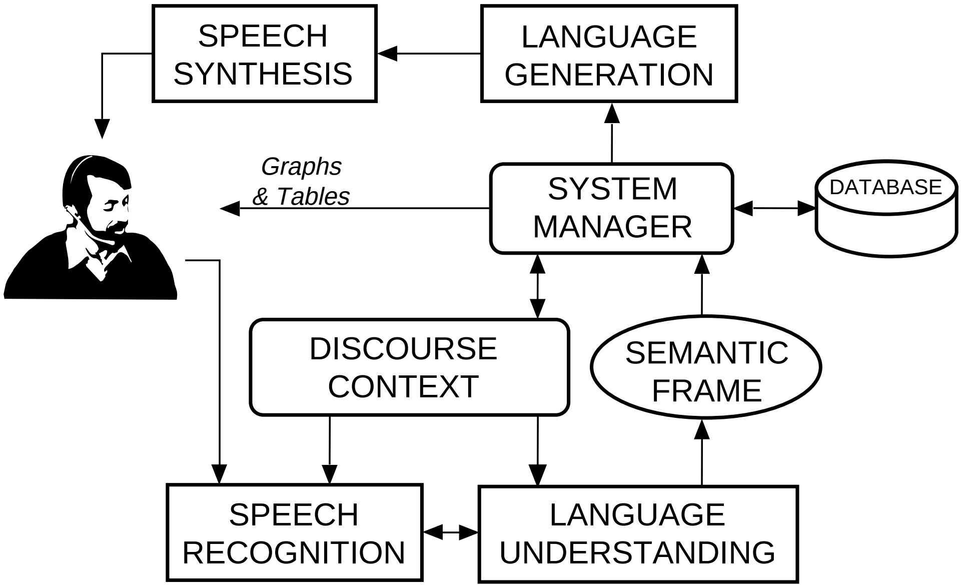
SLS is ten years old!

For many years, the founding members of the Spoken Language Systems (SLS) Group conducted their research at the Speech Communication Group, headed by Prof. Kenneth Stevens, at the Research Laboratory of Electronics. The SLS group was established at the MIT Laboratory for Computer Science in 1989. For readers not familiar with the goals of our research, motivations of our approaches, strategies that we have adopted, and technology and systems that we have developed, I refer you to our web page (www.sls.lcs.mit.edu) for background information.

Briefly stated, the group's goal is to develop human language technologies that will one day lead to interfaces for information creation, access, and management anytime, anywhere, and by anybody. With the increasing demand for information access while un-tethered, speech is fast becoming a desirable, if not required, interface. In our view, this capability calls for a new generation of speech-based interfaces that enable the computer to serve as a conversational partner to the user.

Figure 1 shows the major components of a typical conversational system. The spoken input is first processed through the speech recognition component to convert the acoustic signal into words. The natural language component, working in concert with the recognizer, produces a meaning representation. For information retrieval applications illustrated in this figure, the meaning representation can be used to retrieve the appropriate information in the form of text, tables and graphics. If the information in the input utterance is insufficient, the system may choose to query the user for clarification through dialogue interaction. Speech output can be obtained by processing the information through natural language generation and text-to-speech synthesis. Throughout the process, discourse and contextual information is maintained and fed back to the speech recognition and language understanding components.

Figure 1. A generic block diagram of a typical MIT conversational system.



Since 1989, the SLS group has been involved in the development of human language technologies. For example, our segment-based speech recognition system, SUMMIT, represents a radical departure from the prevailing frame-based HMM systems. Our continuing refinement of SUMMIT has led to phonetic recognition performance that is state of the art. Similarly, our probabilistic natural language understanding system, TINA, was one of the first natural language systems specifically designed for speech, instead of text, input. Its utility has been demonstrated repeatedly in different languages and applications. The technology components that we have developed have been integrated into real applications, so that we can study the interactions of various components and collect useful data on how real users will behave when faced with this new generation of speech-based interfaces. Some of these systems, such as the JUPITER system for weather information, have enjoyed widespread use. Last but not least, we continue to play our educational role, teaching courses and supervising students at MIT and elsewhere. Over the ten-year span, we maintained a student population of 10-15, and have graduated many students, totaling more than 50 degrees.

During this reporting period, we continued our research and development in human language technologies and conversational systems on several fronts. Inspired by work at AT&T, we have re-implemented the lexical access search component of our SUMMIT speech recognition system using weighted finite-state transducers (cf. page 29). This parsimonious framework has led to greater flexibility in utilizing constraint and better performance both in accuracy and speed. We continue to refine ANGIE, a hierarchical, morpho-phonological framework for representing speech and language, incorporating several sub-word layers for speech processing (cf. page 31). ANGIE's utility has been demonstrated in many areas, including duration modelling, speech recognition and understanding, and new word detection and learning.

On another front, we have completed a major redesign of GALAXY a client-server architecture that we introduced in 1994 for accessing on-line information using spoken dialogue. This effort was motivated by the fact that, in 1998, GALAXY was designated as the first reference architecture of the newly launched DARPA Communicator initiative. As a result, we developed a new version of the GALAXY architecture, this time with the specific goals of promoting resource sharing and plug-and-play across multiple sites (cf. page 9). To enable multiple sites to experiment with different domains, components, and control strategies, we made the hub "programmable," i.e., a scripting language controls the flow through each dialogue without recompiling. The hub communicates with the various servers via a standardized frame-based protocol. The resulting GALAXY-II architecture has been delivered to MITRE for distribution, documentation, and maintenance. At the DARPA Communicator's PI meeting in June 1999, eleven organizations demonstrated their GALAXY-II architecturally compliant systems.

With regard to system development, several aspects are particularly noteworthy. First, we continue to develop JUPITER, a telephone-only system for weather information for more than 500 cities worldwide. Since May 1997, we have collected over 250,000 utterances from more than 40,000 calls to a toll-free number. These data have proved to be invaluable for system development, training, and evaluation. In recent months, JUPITER has become our testbed for developing multilingual conversational interfaces (cf. page 17). Following the success of JUPITER, we have started to develop other, similar applications. For example, PEGASUS can provide flight status information (cf. page 19) and VOYAGER can provide traffic and navigation information (cf. page 20). More recently, we have started to develop MERCURY, a system that can help with air travel planning (cf. page 23). Mercury requires considerably more complex dialogue management than the other systems. All these systems provide real information that is updated in a timely manner, and they can all be accessed via a telephone.

With a large number of students completing their degrees the previous year, the SLS student body is in its rebuilding phase. The research topics that students are addressing include acoustic modelling (heterogeneous features, non-native speakers, etc.), word-and phone-level confidence scoring, sub-word modelling for speech recognition and understanding, tone recognition, prosodic modelling, concatenative speech synthesis, analyses and recognition of audio content, speech based document retrieval, and semantics-based information extraction from HTML documents. A brief description of their work can be found in this document. Copies of completed theses, as well as all our publications, are available for download from our web page.

RESEARCH

REPORTS

GALAXY-II

Stephanie Seneff, Raymond Lau, Christine Pao and Nikko Ström

Through our experience over the last decade in designing conversational systems, we have come to realize that an essential element in being able to rapidly configure new systems is to allow as many aspects of the system design as possible to be specifiable without modifying source code. To this end, we recently redesigned our core architecture to support complex system configurations controlled by a run-time executable scripting language. Using this new framework, we have been able to configure multi-modal, multi-domain, multi-user, and multilingual systems with much less effort than previously. We are discovering that we can now configure systems whose capabilities are well beyond what was previously considered feasible.

The resulting new architecture, GALAXY-II [8], has recently been designated as the initial common architecture for the multi-site DARPA Communicator project in the United States. A main goal of this program is to promote resource sharing and plug-and-play interoperability across multiple sites for the research and development of dialogue-based systems. MIT has been given the responsibility of developing the architecture, which is being maintained and distributed from MITRE Bedford[1]. Research on the architecture design is ongoing, and MIT has thus far delivered several updates of the libraries to MITRE. Multiple sites[2] have successfully adopted the framework and incorporated it into their dialogue system research platforms.

GALAXY-II differs from its predecessors mainly in two ways: (1) a central hub handles *all* communications among the various servers via a standardized protocol[3], and (2) system control flow is maintained through a specialized run-time executable programming language interpreted by the hub. GALAXY was first introduced in 1994, as a client-server architecture for accessing on-line information using spoken dialogue [2]. Since then, GALAXY has served as the testbed for our research and development of human language technologies, resulting in systems in different domains (e.g., automobile classified ads [4], restaurant guide [5] and weather information [11]), different languages [9], and different access mechanisms [4,5,11]. In 1996, we made our first significant architectural redesign to permit universal access via any web browser [3]. The resulting WEBGALAXY architecture makes use of a “hub” to mediate between a Java GUI client and various compute and domain servers, dispatching messages among the various servers and maintaining a log of server activities and outputs.

In the process of developing dialogue control modules for various domains in GALAXY, we came to the realization that it is critical to be able to allow researchers to easily visualize program flow through the dialogue, and to flexibly manipulate the decision-making process at the highest level. To this end, we developed a simple high-level scripting language that permits boolean, string, and arithmetic tests on variables for decisions on the execution of particular functions. A domain-dependent dialogue control table specifies a set of sequential rules in this scripting language.

---

[1] For a description of the Communicator Program and documentation of the architecture, see http://fofoca.mitre.org.

[2] Including AT&T, BBN, CMU, SRI, MIT Lincoln Laboratory, MITRE, Lockheed Martin, and HRL.

[3] For high bit rate transmissions the hub brokers the relationship and the data are transmitted directly between the two servers.

Generally, multiple rules fire in the course of a single turn. We found this mechanism to be very powerful, and were successful in incorporating it into our newest domain servers for weather and flight status information[4]. We then began to contemplate the idea of incorporating an analogous mechanism into the program control of the entire system, which was being maintained by the GALAXY hub. At about the same time, discussions were beginning on the possibility that GALAXY be designated as the reference architecture for the DARPA Communicator Program. It seemed possible for a scripting language, modelled after the dialogue tools developed for our domain servers, to support a *programmable* hub for the DARPA Communicator.

In the design of GALAXY-II, we retained the notion of a central hub, but regularized the communication protocol between the hub and all servers, permitting users to configure "hub scripts" to easily specify the flow of information among servers performing their specialized tasks in the course of a dialogue turn. Analogous to the dialogue control table, sequential rules fire based on tests on hub variables. The hub variables are represented in a data structure that we call a "frame," which permits typed variables (e.g., string, integer, float, binary, and (recursively) frame) to be packaged together, manipulated, and transmitted.

An interesting research issue also addressed here is how the complex tasks of mixed-initiative dialogue systems should be partitioned into a set of semi-autonomous servers, each of which has clearly assigned roles. If the community intends to experiment with plug-and-play options, then it will be important to partition the space into servers in a consistent way. It is logical to define separate servers for speech recognition, natural language understanding, natural language generation, and speech synthesis. However, the components that deal with context resolution, response planning, and database retrieval are not necessarily organized the same way by different groups of researchers. In the systems we have thus far designed at MIT, the task of "turn management" is handled by a suite of domain-specific servers, as mentioned previously. Each of these servers is controlled by a separate dialogue control table. However, a single database server takes care of database needs for all of the domain servers, with capabilities of consulting both the Web and local relational databases. The turn managers routinely consult the database multiple times[5] in the course of resolving a single user query. Discourse inheritance is managed separately from turn management, and the context record is updated after both the user turn and the system turn. All domains are handled by a single generic server, but controlled by domain-specific discourse tables [6].

The Hub Scripting Language

The GALAXY-II system consists of a central hub that controls the flow of information among a suite of servers, which may be running on the same machine or at remote locations. The hub interaction with the servers is controlled via a scripting language. A script includes a list of the *servers*, specifying the host, port, and set of operations each server supports, as well as a set of

---

[4] This mechanism is now utilized by all the domain servers in GALAXY-II.

[5] In module-to-module subdialogues.

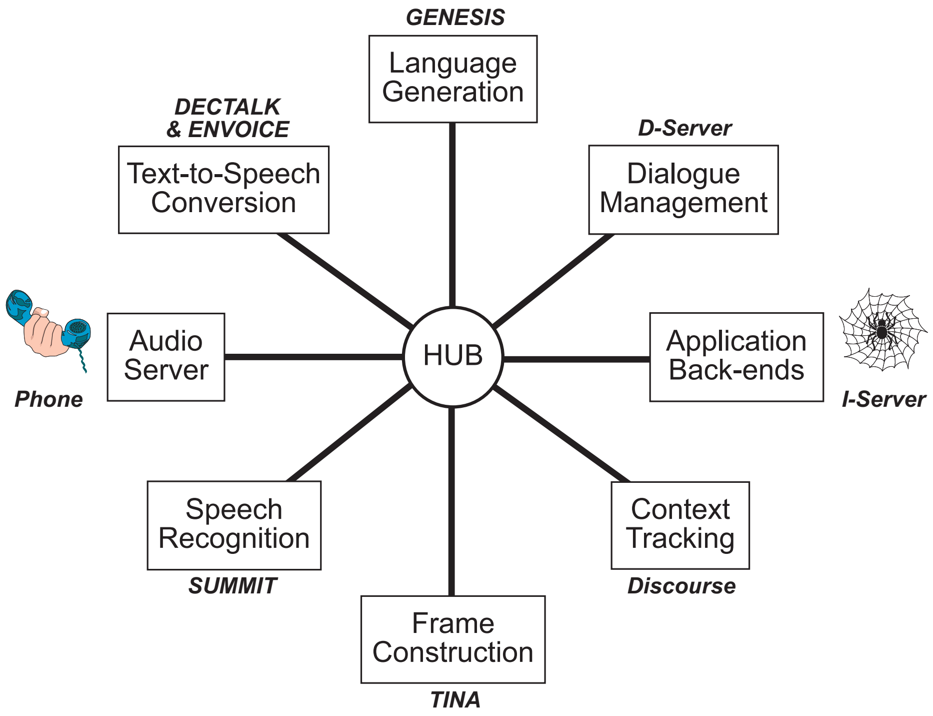


Figure 2. Architecture of GALAXY-II.

one or more *programs*. Each program consists of a set of *rules*, where each rule specifies an *operation*, a set of *conditions* under which that rule should "fire," a list of input and output *variables* for the rule, as well as optional store/retrieve variables into/from the discourse history.

When a rule fires, the input variables are copied into a *token* and sent to the server that handles the operation. The hub expects the server to return a token containing the output variables at a later time. There is the option of no output variables, in which case interaction is one-way only. The input and output variables are all recorded in a hub-internal *master token*. The discourse history will also be updated, if the rule has so specified. The conditions consist of simple logical, string, or arithmetic tests on the values of the typed variables in the master token. The hub communicates with the various servers via a standardized frame-based protocol.

Each individual user is associated with a unique *session*; user state information, such as the current language, domain, etc., is recorded via *session variables*. Each session is usually associated with a particular GUI and/or audio server. Discourse context is organized utterance-by-utterance within a session. Variables can be passed among different tokens associated with the same session via a device of prepending "hub_session" to the key's name. Tokens associated with different sessions compete for available resources, and are queued up by the hub when requested servers are busy. The hub automatically garbage collects tokens when they are no longer active.

Example Rule

An example rule is shown in Figure 3. This rule states that, if a ParseFrame exists but a RequestFrame has not yet been generated

Figure 3. Example rule in the hub script.

```
RULE:         :ParseFrame & !:RequestFrame --> context_tracking
RETRIEVE:     :HistoryFrame
IN:           :ParseFrame
OUT:          :RequestFrame :HistoryFrame :Domain
LOG_OUT:      :RequestFrame :Domain
STORE:        :HistoryFrame
```

(denoted by "!"), then call the "context_tracking" operation, by sending the ParseFrame to the discourse module[6] for evaluation. Also send the previous history to define the context, which is retrieved from the hub's history record logged with the session. This operation, when completed, will return a token to the hub, containing an updated version of the HistoryFrame, a commitment to a particular domain, and a RequestFrame, which is the user query expanded to include any augmentations due to inheritance rules. The updated HistoryFrame is stored in the history record to become context for the next utterance. The RequestFrame and Domain are recorded in the log file.

Program Flow Control

A simple communication protocol has been adopted and standardized for all hub/server interactions. Upon initiation, the hub first handshakes with all of the specified servers, confirming that they are up and running and sending them a "welcome" token that may contain some initialization information, as specified in the hub script. The hub then launches a wait loop in which the servers are continuously polled for any "return" tokens[7]. Each token is named according to its corresponding program in the hub script, and may also contain a rule index to locate its place in program execution[8], and a "token id" to associate it with the appropriate master token in the hub's internal memory. The rule is consulted to determine which "OUT" variables to update in the master, and which variables, if any, to store in the discourse history or the log file. Following this, the master token is evaluated against the complete set of rules subsequent to the rule index, and any rules that pass test conditions are then executed. A top-level flag controls whether the program is to run in "single-threaded" or "multi-threaded" mode, where the former permits only a single rule to fire and the latter immediately executes *all* rules that fire. Servers other than those that implement user-interface functions are typically stateless; any history they may need is sent back to the hub for safekeeping, where it is associated with the current utterance. Common state can thus be shared among multiple servers. Furthermore, state is insensitive to server crashes.

To execute a given rule, a new token is created from the master token, containing only the subset of variables specified in the rule's "IN" variables[9]. This token is then sent to the server assigned to the execution of the operation specified by the rule. If it

---

[6] "Context_tracking" would have been included in the list of operations supported by the discourse server.

[7] Servers can also spontaneously send tokens to the hub without having first received a token from the hub.

[8] The sequential constraints can be easily eliminated, leading to a type of control that resembles an open-agent architecture.

[9] There may also be variables set immediately by a "PARAM" field.

is determined that the designated server is busy (has not yet replied to a preceding rule either within this dialogue or in a competing dialogue) then the token is queued up for later transmission[10]. Thus the hub is in theory never stalled waiting for a server to receive a token. The hub then checks whether the server that sent the token has any tokens in its input queue. If so, it will pop the queue before returning to the wait loop.

Semantic Frame Representation

We expect that researchers utilizing the GALAXY-II system will be developing servers which will need to interface with a suite of existing servers already in place. In such cases, it is necessary for the servers to share a common language in the representations they jointly process. Researchers who choose to replace *all* the servers are free to use whatever meaning representations they find convenient. However, if the intent were to replace a subset of servers, for example, a new dialogue manager or a new language generation server, then the new server would have to adopt the meaning representation protocol that was in use by the replaced component[11]. Thus we think it is appropriate to provide a brief description of the meaning representation formats that have been adopted by our systems.

In the process of developing conversational systems in multiple domains over the last decade, we have constructed a minimal linguistic specification of a meaning representation that we feel is adequate for most applications of interest to us. Our TINA system [7] converts recognizer hypotheses into *semantic frames* in this format, and our context tracking component [6] depends critically upon this format for proper functioning. Our GENESIS [1] system can paraphrase semantic frames into multiple languages, not just natural languages but also into SQL, into a flattened “E-form” representation, and into waveform concatenations for our *Envoice* speech synthesizer [10].

We view the linguistic/semantic world as consisting of three main types of constituents, which we call *clause*, *topic*, and *predicate*[12]. A clause constituent generally occurs at the highest level, and usually represents the high level goal of the user request, which could be, for example, “display,” “record,” “repeat,” “reserve,” etc. Topics generally correspond to noun phrases, and predicates are typically attributes, which could be expressed as verb phrases, prepositional phrases, or adjective phrases. A semantic frame is, then, a named and typed structure, with one of the above three types.

Semantic frames also contain *contents*, and there is a library of tools available for manipulating the contents. Traditional linguistic contents include an optional topic and zero or more predicates. A frame can also contain a set of (key: value) pairs, where the key can be any symbol-string, and the value is one of: (1) an integer, (2) a string, (3) a semantic frame, and (4) a list of values in categories (1)-(3). We use the (key: value) notation for syntactic features such as number and quantification; there is also a distinguished “name” key for named entities. The (key: value) notation is very

---

[10] If multiple instances of a given server are running then it will redirect the query to one that is not currently busy.
[11] Or at least be able to translate it into something more convenient for its purposes.
[12] A somewhat different intent from the usual definitions of these terms.

Figure 4. Example semantic frame for the sentence, "Show me flights from Boston to Denver."

```
clause:
  { display
    topic:
        { flight
          number: pl
          predicate:
               { from
                 topic: { city name: Boston }
               }
          predicate:
               { to
                 topic: { city name: Denver }
               }
         }
  }
```

generic, and it has allowed us to represent almost any information we need to record, most especially database retrievals, in semantic frame format. For instance, the key "airline" has the value "United" as retrieved from the database. In fact, the token that is sent between the hub and the servers is also itself a [degenerate] instance of a semantic frame[13], although at the highest level it only utilizes the (key: value) feature of the frame.

An example semantic frame for the sentence "Show me flights from Boston to Denver." is shown in Figure 4.

Audio Server

The audio server has been upgraded to handle multiple simultaneous user sessions. The new server has a well-defined interface to computer telephony (CT) hardware, which can be implemented for different CT APIs. However, we are currently using hardware from Dialogic Corp. With this new server, multiple phone lines can be connected to a system, or different systems may share lines from one audio server.

The new server has an event-based design that allows it to react faster to user input and system responses. Furthermore, by continuously playing and recording audio on the hardware, we gain total software control of the audio. A useful side effect is that entire sessions can be continuously recorded, and therefore effects of speech detection errors and user barge-in can more easily be analyzed.

To further increase the system's responsiveness to user input we have improved the speech detection algorithm. We no longer rely on simply the signal energy to make a speech/non-speech decision, but we also take the degree of periodicity into account. The improved accuracy is used to reduce the speech-end detection time. We have also improved the logic for generating so called horizon events. A horizon event is a signal that tells the recipient that no events will occur before a certain time. By sending appropriate horizon events to the recognizer we optimize the CPU resources spent in real time by the recognizer.

---

[13] It is a clause whose name is the program it refers to.

Future Plans

Over the past year, we invested significant resources towards the development of the GALAXY-II architecture, but we feel that the result was well worthwhile. We have generally found that the flexibility inherent in hub scripting empowers us to conceive of systems that we formerly considered to be impossibly complex. We have recently envisioned two extensions to our GALAXY-II system that we feel will be relatively straightforward to implement.

The first would be a wizard-mode system that makes use of two telephone interfaces, one of which records input from the user and speaks system responses to the same user, the other of which plays these same user queries to a wizard who then speaks a "translation" of the user query, either in a simplified form in the same language, or into a different language that the system understands. The wizard's query gets processed through usual channels, except that the audio output is directed to the user, and the GUI output is displayed on the wizard's screen. We believe that such a setup, which will be a powerful mode for collecting user data for maturing systems, can be configured completely within the hub script.

A second mode that we are just beginning to explore is a system that behaves as an *agent*, calling back a user when a predesignated condition is met. We envision an "agent" server, which might be monitoring a number of events, such as a particular flight or the traffic on a given highway. As soon as the specified condition has occurred (the flight has arrived, the traffic is at a standstill, etc.) the agent server issues a "call me" request, providing both the appropriate phone number and the appropriate "welcome" message, detailing the outcome of the event. Our initial efforts in developing a system supporting such capabilities have been very encouraging. We have been able to rapidly configure hub scripts to support interaction between the agent server and other domain servers (to field questions) and between the agent server and the user (both for calling them up at the appropriate time and for negotiating a new agent activity to be performed in the future).

References

[1] J. Glass, J. Polifroni and S. Seneff, "Multilingual Language Generation Across Multiple Domains," *Proc. ICSLP '94*, pp. 983–986, Yokohama, Japan, Sept. 1994.

[2] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue, "GALAXY: A Human Language Interface to Online Travel Information," *Proc. ICSLP '94*, pp. 707–710, Yokohama, Japan, Sept. 1994.

[3] R. Lau, G. Flammia, C. Pao, and V. Zue, "WEBGALAXY - Integrating Spoken Language and Hypertext Navigation," *Proc. Eurospeech '97*, pp. 883–886, Rhodes, Greece, Sept. 1997.

[4] H. Meng, S. Busayapongchai, J. Glass, D. Goddeau, L. Hetherington, E. Hurley, C. Pao. J. Polifroni, S. Seneff, and V. Zue, "WHEELS: A Conversational System in the Automobile Classifieds Domain," *Proc. ICSLP '96*, Philadelphia, PA, pp. 542-545, Oct. 1996.

[5] S. Seneff and J. Polifroni, "A New Restaurant Guide Conversational System: Issues in Rapid Prototyping for Specialized Domains," *Proc. ICSLP '96*, pp. 665-668, Philadelphia, PA, Oct. 1996.

[6] S. Seneff, D. Goddeau, C. Pao, and J. Polifroni, "Multimodal Discourse Modelling in a Multi-User Multi-Domain Environment," *Proc. ICSLP '96*, pp. 192–195, Philadelphia, PA, Oct. 1996.

[7] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, pp. 61–86, 1992.

References (continued)

[8] S. Seneff, P. Schmid, E. Hurley, R. Lau, C. Pao, and V. Zue, "GALAXY-II: A Reference Architecture for Conversational System Development," *Proc. ICSLP '98*, pp. 931-934, Sydney, Australia.

[9] C. Wang, J. Glass, H. Meng, J. Polifroni, S. Seneff, and V. Zue, "YINHE: A Mandarin Chinese Version of the GALAXY System," *Proc. Eurospeech* '97, pp. 351—354, Rhodes, Greece, Sept. 1997.

[10] J. R. Yi and J. R. Glass, "Natural-sounding Speech Synthesis using Variable-length Units," *Proc. ICSLP '98*, pp. 1167-1170 Sydney, Australia, Nov. 1998.

[11] V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid, "From Interface to Content: Translingual Access and Delivery of On-Line Information," *Proc. Eurospeech* '97, pp. 2227—2230, Rhodes, Greece, Sept. 1997.

Multilingual JUPITER

Stephanie Seneff, Joseph Poliforni, Yasuhiro Minami and Yi-Chung Lin

We have recently begun work on porting the JUPITER system to a number of other languages, initially concentrating on Japanese and Chinese. These efforts are taking place within the context of the GALAXY-II architecture (c.f. page 53). Because the information that JUPITER maintains is represented in a semantic frame format, it is straightforward to translate this information to another language besides English. We have begun an effort to paraphrase the weather responses into Japanese, Mandarin Chinese, German and Spanish. For each of these languages, a native speaker who is fluent in English is preparing the corresponding GENESIS generation rules.

We had not dealt extensively in the past with German, a particularly difficult language due to its extensive use of inflectional endings. We had to augment GENESIS with a more sophisticated ability to deal with case, which can be assigned in the vocabulary file by prepositions and verbs. In addition, we needed to be able to specify the correct inflectional endings for nouns and adjectives as a function of case, gender, and number.

There were a few instances when the same word in English had to be given a different translation depending on the context. For example, the word “light,” translates differently into Mandarin for the two phrases, “light wind” (“qinq1”) and “light rain” (“xiao3”). GENESIS handles this situation using a semantic grammar that can categorize the two cases into different adjective types.

Figure 5 gives an example of a semantic frame for the sentence, “2 to 4 inches snowfall accumulation by morning,” along with the corresponding paraphrases in three languages. Note that the preposition “by” has been interpreted in the semantic frame as denoting a time expression, allowing the appropriate translation of this diversely realized preposition.

With the help of visiting researchers, we are working on the paraphrasing of incoming English weather reports into Chinese and Japanese. Because local sources provide the most detailed weather information, we also plan to obtain weather reports from Japanese and Chinese information providers. The data from these local sources can be paraphrased into English so that our

Figure 5. Example semantic frame for the sentence “two to four inches snow fall accumulation by morning” with paraphrases in three languages.

```
clause: weather_event
   topic: accumulation
       name: snowfall
       pred: amount
         topic value, name: 2
            pred: to_value
              topic: value, name: 4, units: inches
       pred: by_time
           pred: time_interval
             topic: time_of_day, name: morning

Input:    2 to 4 inches snowfall accumulation by morning
English:  snowfall 2 to 4 inches by morning
German:   Schneefall 2 bis 4 Inch bis am Morgen
Spanish:  nevada 2 a 4 pulgadas antes de la manana
```

English system can benefit as well from this more in-depth knowledge.

We have had Chinese recognition, understanding, and generation capability for many years and we recently incorporated those capabilities into the current GALAXY architecture. It is now possible for a user to speak to the system in Chinese, see the response in Chinese characters and hear the response in English. Although we do this because we do not yet have a Chinese speech synthesizer, it also demonstrated the strength and flexibility of the hub architecture, in allowing us to specify separate grammars and generation catalogs within a given discourse context.

PEGASUS

Stephanie Seneff and Joseph Polifroni

Our PEGASUS flight status system was first introduced in 1998 [1]. Over the past year we have paid particular attention to database issues in PEGASUS. Our database of flight status information, which is updated every six minutes, is now kept current on flights throughout the entire calendar day. Previously, as flights disappeared from our updates (approximately two hours after landing), they were also expunged from our database. Now users can find on-time information for any flight that has taken off and/or landed on the particular day they ask. Flight records in our database now maintain a timestamp for their latest update, which the PEGASUS system consults when deciding on how to answer a query. Only current information (defined as information obtained that same day) is provided to the user.

We are currently exploring a new mechanism for updating the database of schedule information, which the PEGASUS system uses to answer queries that do not contain an airline and flight number. We have found it important that this information be kept as current as possible. Therefore, flight schedules are now obtained live at the time of the user query via an on-line information provider.

We have developed more sophisticated mechanisms for dealing with errors caused by misrecognition of flight numbers. We were able to implement fairly complex algorithms quickly, directly as a consequence of the hub scripting capabilities of GALAXY-II. When the system proposes a query containing an airline and flight number that is not in our schedule or status databases, the N-best list is set aside while the system engages the user in a clarification sub-dialogue to elicit a source and destination city. The system then queries the database for a set of flight numbers consistent with those two cities and refilters the N-best list based on these new flight numbers. After this short sub-dialogue and using this intelligence, the system is able to correct a flight-number misrecognition without ever revealing to the user that a mistake was made.

Reference

[1] S. Seneff, J. Polifroni and P. Schmid, "PEGASUS: Flight Departure/Arrival/Gate Information System," *MIT Spoken Language Systems Group Research Summary*, p. 25-26, July 1998.

Development of the VOYAGER System

TJ Hazen

Over the course of the last year, a significant effort has been placed on the development of a new and improved VOYAGER system. VOYAGER has been one of the cornerstone systems of the Spoken Language Systems group since its inception in 1989 [1,2]. The basic function of the system is to provide tourist and travel information for the city of Boston. In the past, the system has focused on static map-based information, allowing the user to ask about sites and landmarks, view maps, and obtain directions from place to place.

In the last year the VOYAGER system has undergone a major overhaul in two main areas. First, the system has been completely ported to the new GALAXY-II spoken language system architecture [3]. Second, it has been augmented with new, dynamic information about current traffic conditions (as provided by SmartRoute Systems). These two changes have allowed the system to become viable for use by the general public over a standard telephone line.

In developing the new VOYAGER system, various technical challenges have presented themselves. First, the complexity of the queries that can be asked of VOYAGER exceeds the complexity of queries typically asked of our other systems, such as JUPITER [4]. This required that the flat key-value dialogue representation used in our previous systems be expanded to handle lists and hierarchical constructs. With this augmented dialogue structure complex requests, such as “Show me a map with museums and historic sites in Boston, Brookline and Cambridge”, can be handled. From the query the system generates the semantic representation shown in Figure 6, for use by the dialogue manager and database server.

The inclusion of a hierarchical structure and lists in this query allows for a straightforward representation of the complex query which, despite its increased complexity, is easily handled by the back-end components of the system (i.e., the dialogue manager and the database server).

The particular query also contains an example of semantic ambiguity, which the turn manager is responsible for resolving. The hierarchical structure indicates that the parser interpreted the query as a request for “museums” first and then “historic sites in Boston, Brookline, and Cambridge” second. Although this particular parse is technically an acceptable interpretation, the more

Figure 6. Eform structure associated with the query “show me a map with maps and historic sites in Boston, Brookline and Cambridge”.

```
{c eform
    :action “display”
    :list ( {q list_item
               :category “museum” }
            {q list_item
               :category “historical_site”
               :in ( {q list_item
                        :category “city”
                        :name “Boston” }
                     {q list_item
                        :category “city”
                        :name “Brookline” }
                     {q list_item
                        :category “city”
                        :name “Cambridge” } ) } ) }
```

appropriate interpretation would attach the "in Boston, Brookline, and Cambridge" prepositional phrase to both "museums and historic sites". Currently, the turn manager has a set of heuristic rules for detecting and correcting these types of parse ambiguities.

Another technical challenge of VOYAGER is the ability to generate short and concise traffic reports from the traffic information reports provided by SmartRoute. The traffic reports are distributed in small capsules, each reporting the traffic conditions for a particular roadway segment. Typical users will ask for a traffic report covering multiple roadway segments. The challenge is to aggregate the traffic reports from these roadway segments without generating excessive or redundant information. To provide an example, consider the reports generated for the following user query, "What is the traffic like on Route 2?" The database returns reports for four different segments of road for this query. If the four reports are generated independently then the user might hear the following :

```
On Route 2 heading east from
Route 495 to Route 95 the
traffic is moving freely with an
average speed of 41 miles per
hour. On Route 2, heading east
from Route 95 to Memorial Drive,
the traffic is moving freely
with an average speed of 36
miles per hour. From the Ground
Round Rotary to Huron Avenue,
there is a utility crew. Various
lanes are restricted. You are
advised to expect delays. On
Route 2 heading west from
Memorial Drive to Route 95 the
traffic is moving freely with an
average speed of 36 miles per
hour. On Route 2 heading west
from Route 95 to Route 495, the
traffic is moving freely with an
average speed of 46 miles per
hour.
```

However, the system contains a series of rules designed to cut down on the verbosity of the traffic reports that are generated. These rules can apply various linguistic phenomena, such as segregatory coordination and ellipsis, as well as some task specific heuristics for eliminating redundant information. After applying these rules, the report above is reduced to the more concise report listed below:

```
On Route 2 heading east from
Route 95 to Memorial Drive, the
traffic is moving freely with an
average speed of 36 miles per
hour. From the Ground Round
Rotary to Huron Avenue, there is
a utility crew. Various lanes
are restricted. You are advised
to expect delays. Traffic on the
other roadway segments for Route
2 is moving freely at or near
the speed limit.
```

Various other research areas that have been investigated under VOYAGER include the development of a robust recognizer trained from out-of-domain data and the study of the dialogue management issues surrounding reading driving directions to a user over the phone.

In the coming year, we hope to continue to improve the VOYAGER system. One potential future direction for VOYAGER is its use in a mobile environment. If VOYAGER becomes portable and is augmented with GPS information, it can be used in a mobile environment such as an automobile. This will allow the system to dynamically update the user about travel conditions or information based on the user's current position as well as the user's stated goals and desires.

References

[1] V. Zue, "Toward Systems that Understand Spoken Language," IEEE *Expert*, pp. 51-59, February, 1994.

[2] J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue, "Multilingual Spoken-Language Understanding in the MIT VOYAGER System," *Speech Communication*, Vol. 17, No. 1, pp. 1-18, March 1995.

[3] S. Seneff, R. Lau, and J. Polifroni, "Organization, Communication, and Control in the GALAXY-II Conversational System," *Proc. Eurospeech '99* Budapest, Hungary, September, 1999.

[4] V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid, "From Interface to Content: Translingual Access and Delivery of On-Line Information, " *Proc. Eurospeech* '97, pp. 2227-2230, Rhodes, Greece, September, 1997.

The Mercury System for Flight Browsing and Pricing

Stephanie Seneff, Raymond Lau, James Glass and Joseph Polifroni

In June, 1999, we completed the first working demonstration of our MERCURY system for obtaining flight schedule information and for pricing potential itineraries via an over-the-telephone conversational interface. Mercury supports queries about air travel amongst the 150 busiest airports worldwide, as measured by 1997 passenger traffic, and can help a caller plan and price an intinerary involving one or more destinations. Information for MERCURY is presently obtained from The Sabre Group's Travelocity service. In June, we conducted an initial evaluation of the system involving eighteen callers from within our group but who were not system developers. On average, thirty-six queries were required to plan an itinerary involving three legs. The average call duration was thirteen minutes.

User Interaction Paradigm

The current MERCURY system presumes that the user is a schedule-sensitive, as opposed to price-sensitive, traveler attempting to plan an air itinerary. The system supports both enrolled users and guest users. For enrolled users, certain profile information, such as initial origin city, is assumed and the user is provided with an option to receive an emailed copy of his/her itinerary at the end. The user is asked to construct his/her itinerary, one leg at a time, making a selection of specific flights for each leg. At any point in time, the user is permitted to modify previous legs by specifying updated constraints. Examples of permissible queries are given in Figure 7. After constructing an itinerary, the user is permitted to ask for a price, and if enrolled, to have the final itinerary, with pricing information, emailed to him/her. An example of a complete interaction with the system is shown in Figure 8.

System Architecture

The MERCURY system is built within the GALAXY-II architecture ([1]). MIT's SUMMIT ([2]) is used for the speech recognition component, TINA ([3]) is used for natural language understanding and context-tracking, GENESIS ([4]) is used for language generation. For speech synthesis, a commercial-off-the-shelf solution (DECTalk) is used. The telephony interface is a Dialogic board. The entire system runs in real time primarily on Intel-based PCs. Information on flight schedules and pricing is obtained in

Figure 7. Examples of permissible queries for MERCURY along with some comments indicating supported functionality.

```
I want to fly from Boston to London on British Air next Friday.
      [Interpretation of dates.]

I want to depart in the morning.
       [Adding a departure time constraint to present leg.]

I want to arrive in Tokyo a little earlier.
       [Modification of constraints for a leg to Tokyo already
       in partially constructed itinerary.]

I want to return three days later.
       [Date is relative to arrival date on previous leg, and
       destination is inferred to be origin of itinerary.]
```

real-time from the Travelocity Website via screen-scraping. A small local database of pragmatic information, such as geographic location of airports, provides supplemental input into the turn manager component.

Dialogue Challenges

We believe the flight domain to be much more complex, in terms of potential dialogue interactions, than domains of our other demonstration systems, such as the JUPITER weather information system. Planning an air travel itinerary involves more than basic information access. The traveler, together with the system, must jointly arrive at a solution set that satisfies both the constraints of the traveler and those inherent in the availability of commercial flights. The natural expression of these constraints also typically involve numerous pragmatic details, such as the structure of the calendar, time zone differences, geographical proximity of airports, etc. Beyond the challenges of understanding and negotiating constraints, a dialogue typically also involves the user wishing to browse multiple flight choices suggested by the system. In such cases, inexact references to an item in a list must be understood by the system. For example, a flight departing at 7:56 AM might be referred to as "the eight o'clock flight." In order to properly handle such references, we developed a "fuzzy" matching heuristic that accounts for departure/arrival times, airline names, flight numbers, and source/destination/connection cities. Another complication is that typically, there are two lists to which a user can refer - the partially constructed itinerary and the list of recently retrieved flights. Contextual information, along with heuristics accounting for the pragmatics of competing references, need to be employed.

An extremely ambiguous aspect of air travel planning is that users have numerous ways of expressing an intention to book a flight but not all such manners of expression are necessarily booking requests. For example, if the user says "the later flight," it is unclear whether he wants more information or wants an actual booking. MERCURY adopts the general strategy of treating these potential booking requests as more information requests and follows up with "Shall I book this flight for you?" A similar confirmation strategy is employed by MERCURY when "dangerous" actions are requested by the user. For example, we discovered that when city name requests are misrecognized and the source city is changed during the itinerary planning stage, an extensive repair process must be undertaken by the user to recover. To help protect against this, the system now explicitly requests confirmations in such instances. The top level dialogue control table for the initial demonstration system consists of approximately 200 if-then rules.

Response Challenges

There are numerous challenges in generating a response to a user query, particularly when a display is not available. Typically, numerous flights are available to satisfy a given set of user constraints. This list must be prioritized to one to three reasonable choices to allow the system to present information in fragments small enough not to be overwhelming. A heuristic accounting for non-stop, time and airline preferences is used for the prioritization. During the negotiation between the user and the system to arrive at a set of mutually agreeable constraints, we would like the system to

```
MERCURY: Welcome to MERCURY, the MIT air travel planning system.
         I can help plan your air travel itinerary. Please tell
         me your user name. If you do not have a user name,
         please say new user.

USER:    NEW USER

MERCURY: Welcome. You have successfully logged onto MERCURY as a
         new user. How may I help you?

USER:    I WOULD LIKE TO FLY FROM BOSTON TO LONDON

MERCURY: Flights from Boston to London. What date will you be
         traveling?

USER:    NEXT FRIDAY IN THE EVENING

MERCURY: Okay. flights from Boston to London departing between
         5:00 pm and midnight o n June 25. Can you provide a
         departure or arrival time? The earliest nonstop flight
         from Boston to London departing between 5:00 pm and
         midnight on June 25 leaves at 5:30 p.m. and arrives at
         4:55 a.m. on British Airways. You arrive one day later.

USER:    IS THERE A UNITED FLIGHT?

MERCURY: Okay. United flights from Boston to London departing
         between 5:00 pm and mid night on June 25. One moment
         please... I have a nonstop United flight leaving at
         6:40 p.m. and arriving at 6:25 a.m. the following day?

USER:    BOOK THAT FLIGHT

MERCURY: United flight 998 has been added to your itinerary. You
         depart Boston at 6:40 pm and arri ve in London Heathrow
         at 6:25 am. You arrive one day later. Where would you
         like to go next?

USER:    I WOULD LIKE TO RETURN TO J F K THE FOLLOWING TUESDAY

MERCURY: Okay. United flights from London Heathrow to New York’s
         Kennedy on June 29. One moment please... I have a
         flight departing at 1:55 p.m., a flight departing at
         9:00 a.m., and a flight dep arting at 6:00 p.m. Please
         select one.

USER:    BOOK THE EARLIEST FLIGHT

MERCURY: United flight 957 has been added to your itinerary. You
         depart London Heathrow at 9:00 am and arrive in New
         York’s Kennedy at 11:50 a.m.. Where would you like to
         go next?
```

Cont’d

Figure 8. Example interaction with MERCURY.

Figure 8. (cont'd)

```
USER:    I WOULD LIKE TO RETURN TO BOSTON THAT SAME EVENING

MERCURY: Okay. United flights from New York’s Kennedy to Boston
         departing between 5:0 0 pm and midnight on June 29. One
         moment please... I have a flight departing at 8:55 p.m.,
         and a flight departing at 7:00 p.m. Please select one.

USER:    I WOULD LIKE THE SEVEN O+CLOCK FLIGHT

MERCURY: United Airlines flight 7282 leaves New York’s Kennedy at
         7:00 p.m. and arrives in Boston at 8:31 p.m. The total
         flight time is 1 hours and 31 minutes. The aircraft is a
         Jetstream 41 Turbo. Shall I book this flight for you?

USER:    YES PLEASE

MERCURY: United flight 7282 has been added to your itinerary. You
         depart New York’s Kennedy at 7:0 0 pm and arrive in
         Boston at 8:31 pm. Shall I price your itinerary?

USER:    YES

MERCURY: The price for your itinerary is $2065.40. Is there
         anything else I can do for you?

USER:    NO THANK YOU GOOD BYE

MERCURY:Thanks for using MERCURY. Have a nice day!
```

present confirmations to the user of the current desired constraint understood by the system. However, this confirmation soon becomes wordy and inefficient. We have adopted a tersification heuristic which tries to only respond with the change in the constraint set, except when a major milestone, such as the booking of a flight, is reached. Finally, because computerized reservation systems, the ultimate data sources underlying MERCURY, are relatively slow at providing responses, we introduced intermediate replies to inform the user that the query is being processed. When an intermediate reply is used, MERCURY includes the confirmation information as part of the intermediate reply in order to make efficient use of otherwise idle time.

Evaluation

We conducted an initial evaluation of MERCURY on eighteen members of our research group who were not system developers. Of the subjects, five were non-native speakers of English and four were novices on our systems, in that they joined the group recently. The subjects were asked to plan the following itinerary: City A to B on a date in the near future. City B to C on a date n days later where n is at least one. Finally, returning to city A later that same day. A total of 648 utterances were evaluated. The average utterance contained 5.6 words. Each session lasted an average of 12.4 minutes. All but one session lasted under eighteen minutes. The fastest sessions were approximately seven minutes. Figure 9

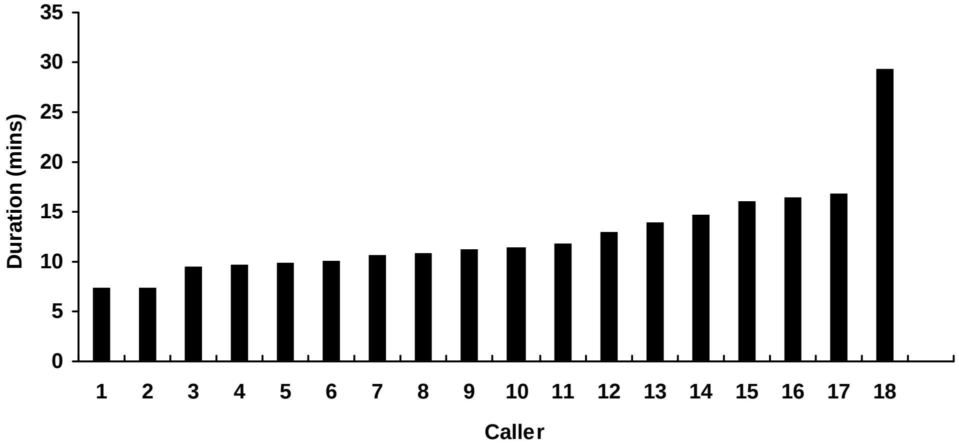


Figure 9. Durations of the eighteen evaluation calls to MERCURY.

shows the distribution of call durations graphically.

All eighteen callers were able to complete an itinerary satisfying the parameters of the task. It is now known how many actually completed a desired itinerary vs. any itinerary. The class bigram language model achieved a perplexity of 20.7 and the class trigram model achieved 17.4. The language models were trained on over 48,000 sentences, drawn from an internal data collection effort and from previous data used in our earlier work in air travel and other domains. The vocabulary size was 959 words. The recognizer, using diphone boundary acoustic models trained on data from our JUPITER weather information system, had an overall word error rate of 20.5% and an overall sentence error rate of 43.7%. Of the sentences, 91% had references and hypotheses that were parseable by the natural language component and thus can be used for natural language understanding evaluation. The remainder consists of 7% where the reference cannot be parsed and 2% where only the recognizer hypothesis cannot be parsed. Various error rates for each of these classes are given in Table 1.

Summary and Future Work

While we have made significant progress on MERCURY as an air travel planning system, several major obstacles remain. One is that prior user-enrollment via an out-of-band mechanism is required. We would like users to be able to enroll directly over-the-telephone or to provide an email address for an itinerary at the end of a session, even without any enrollment. To enable such functionality, we are actively pursuing proper noun and spelling recognition. Another limitation lies in our choice for a user-interaction model - namely having a scheduling-sensitive focus. Our experience with developing conversational systems is that they typically improve only with the collection of more user data from a large user population. However, before we can obtain such user data, we need to convince users to call. While business travelers are scheduling-conscious, that segment of the population will likely continue to deal with their existing travel agent, especially since we are unable to actually issue the final tickets. However, we believe that if instead, we were to provide a service for the price-sensitive shopper, and we are able to deliver

Table 1. Performance statistics for preliminary MERCURY *evaluation. The understanding error rate measures key-value errors.*

	Evaluable	Ref Fails to Parse	Hyp Fails to Parse
# Utterances	587	45	16
WER (%)	13.9	42.1	41.2
UER (%)	20.9	N/A	N/A

competitive pricing and comprehensive ticketing information, such as fare basis codes, that can be used to complete a reservation with an actual travel agent, we will deliver significant enough value to attract users. The primary reason we chose to start with a schedule-sensitive paradigm is that the dialogue interaction is likely to be simpler, because the system needs to worry primarily about one leg at a time. In a price-sensitive paradigm, the user must be encourage to specify a minimal set of constraints to enable effective price searching. The legs of a trip must not be fixed until all user constraints are known. This can greatly complicate the dialogue, but is a challenge we are ready to address. Also, effective price search requires a much higher bandwidth access to a flight pricing database than possible with screen-scraping a Web-based service. We are presently evaluating alternative data providers. Another area we would like to improve is to make the system output sound more natural to the user. Because of the complexity in flight descriptions, improving intelligibility of the system response is important. We believe that by using our ENVOICE ([5]) concatenative synthesizer, we can have much better sounding and more easily understood output.

References

[1] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "GALAXY-II: A Reference Architecture for Conversational System Development," *Proc. ICSLP '98*, pp. 961-964, Sydney, Australia, November 1998

[2] J. Glass, J. Chang, and M. McCandless, "A Probabilistic Framework for Feature-based Speech Recognition," *Proc. ICSLP '96*, pp. 2277-2280, Philadelphia, PA, 1996.

[3] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, pp. 61-86, 1992.

[4] J. Glass, J. Polifroni, and S. Seneff, "Multilingual Language Generation Across Multiple Domains," *Proc. ICSLP '94*, pp. 983-986, Yokohama, Japan, 1994.

[5] J. Yi and J. Glass, "Natural-Sounding Speech Synthesis UsingVariable-Length Units," *Proc. ICSLP '98*, pp. 1167-1170, Sydney, Australia, 1998.

Finite-State Transducers

Lee Hetherington

Spoken language systems make use of a diverse set of models and constraints in translating from spoken utterances to meaning representation and back again. Many of these models and constraints take the form of converting from one level of representation to another. Many of these mappings can be modeled using various forms of finite-state transducers (FSTs) [1,2]. Often these mappings are weighted, meaning that conversions have scores (often probabilities) attached to them. For example, a dictionary or lexicon can represent the conversion or mapping from phonemes to words and vice-versa. Figure 10 is an example showing a small subset of a lexicon.

Furthermore, FSTs representing individual mappings can be chained together, or composed, to create another FST representing the total chain of mappings. FSTs can be combined using composition, concatenation, union, and closure operations to construct new FSTs. FSTs can also be manipulated using identity operations such as epsilon removal, determinization, and minimization to produce equivalent FSTs that may have desirable performance properties.

A big advantage to using FSTs to represent various constraints or models within a spoken language system is that by choosing a common representation, we are able to utilize powerful generic finite-state algorithms at all levels. If components of a system are written to operate on a generic FST, then those components will work on any we provide without having to rewrite code. For example, many speech recognition systems are hardwired to use a bigram language model in the first (Viterbi) pass. However, if this search component is instead written to operate on any FST, then it will also work on arbitrary n-gram, finite-state, or even context-free language models. Furthermore, many of the aforementioned FST operations can be applied on-the-fly, meaning that FSTs do not need to be combined in advance. Thus, an FST that a system component sees may in fact be composed of several FSTs, yet that component need not be aware of this construction.

Over the past two years, we have developed an extensive C++ library and associated tools that implement the various FST operations, allowing us to reimplement the lexical search component of the SUMMIT speech recognition system to utilize FSTs [3]. In particular, the search component sees a single FST encompassing everything from context-dependent acoustic model labels to word-level language constraint. We typically use the cascade of FSTs $C \circ P \circ L \circ G$ as the constraint. Starting at the highest level and working down, we have G representing the grammar or language model constraining word sequences, L the lexicon or dictionary mapping from words to phonemes, P the context-dependent phonological rules mapping from phonemes to phones, and C the mapping from context-independent phones to context-dependent acoustic model labels.

In our system, the language model G can currently be an arbitrary n-gram, finite-state network, or context-free grammar. (Even though a context-free grammar is in general infinite-state, on a finite-length utterance, the search will only explore a finite part of it, so it can appear to be finite when expanded on-the-fly.) Note that G need not simply constrain the word sequences; since it is a transducer, it could be constructed to output some level of

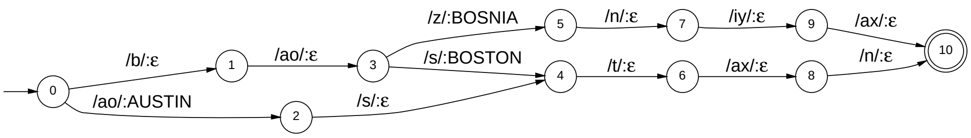


Figure 10. A subset of a lexicon represented as an FST. Transition labels are of the form input:output, with epsilon representing an empty label. In this case, the input labels are phonemes, and the output labels words.

meaning extraction. Thus, for some domains, it may possible for SUMMIT to go from waveform to meaning representation without the need for a more powerful natural language understanding system such as TINA.

Not only can many of the spoken language constraints be represented using FSTs, but many of the system's internal and external results can be represented using them as well. For example, speech recognition output may include N-best paths or word and phonetic graphs (lattices). Our system also represents these using generic FSTs, allowing us to manipulate them using our FST tools and also to use them as constraints for subsequent recognition runs.

Overall, our move to finite-state transducers has really transformed our system. FSTs give us tremendous flexibility in how we configure our system. On one hand, by composing and optimizing FSTs ahead of time we can make SUMMIT both faster and more accurate than previously possible. On the other hand, the system has become much more flexible because we can use arbitrary FSTs that can be combined in novel ways.

References

[1] F. Pereira and M. Riley, "Speech Recognition by Composition of Weighted Finite Automata," in Finite-State Language Processing, E. Roche and Y. Schabes, Eds., pp. 431-453. MIT Press, Cambridge, MA, 1997.

[2] M. Mohri, M. Riley, D. Hindle, A. Ljolje, and F. Pereira, "Full Expansion of Context-dependent Networks in Large Vocabulary Speech Recognition," *Proc. ICASSP '98*, pp. 665-668, Seattle, WA, 1998.

[3] J. Glass, T.J. Hazen, and L. Hetherington, "Real-time Telephone-based Speech Recognition in the Jupiter Domain," *Proc. ICASSP '99*, pp. 61-64 Phoenix, AZ, 1999.

ANGIE

Stephanie Seneff

Over the past several years, we have been exploring various models for accounting for subword structure in American English within a probabilistic framework. The system we have developed, called ANGIE, has been applied thus far to flexible vocabulary speech recognition [2,3,4], hierarchical duration modelling [1], and a reversible letter-to-sound sound-to-letter system [5]. Recently we have used ANGIE to encode English words into a special two-tiered lexicon, wherein words are represented as sequences of "morphs," which in turn are transcribed in an idealized phonemic form. The morphs are analogous to Chinese characters - they are generally syllable-sized units spelled according to the subspellings of the words that contain them, and marked for position and stress.

By leveraging off of a pre-existing 11,000 word lexicon, we were able to obtain a high quality lexicon for over 55,000 words from Pronlex. This was done using a multi-step procedure that began with training an ANGIE grammar on the 11,000 word subset, for both the letter inputs and the Pronlex phone inputs. We then processed each word in the larger lexicon through the following set of steps:

1) Parse the Pronlex phones into a phoneme graph, using ANGIE.

2) Parse the letter spelling into a phoneme graph, restricted to be a subset of the graph obtained in (1).

3) Parse both the dual-parse phoneme graph and the letter spelling into morph graphs based on the morph lexicon, generalized to allow all combinations of onset-rhyme in stressed roots.

4) Compose the two morph graphs into an intersection space, which is then parsed using our TINA natural language system into candidate whole-word solutions in terms of morphs (decomposed into onset-rhyme in the case of stressed roots).

5) If appropriate, propose novel morphs through observed recombination of onsets and rhymes.

Having obtained morph representations for the large lexicon, we can then retrain the ANGIE grammar to improve the quality of the letter-to-sound system. The morph lexicon also serves as a potential vocabulary for a first stage recognizer in a two-stage system, whose intent is to provide linguistic constraint in the absence of a word lexicon. In such a model, the second stage would reparse a high quality phone or letter graph into proposed word hypotheses, utilizing multiple language models (including full parsing support from TINA) in parallel. Each language model would be specific to a particular domain such as weather or flights. Ideally, this would permit seamless domain switching, as well as an ability to handle partial words (false starts) and new or out-of-domain words (cf. page 37).

References

[1] G.Y. Chung and S. Seneff, "A Hierarchical Duration Model for Speech Recognition based on the ANGIE Framework," *Speech Communication*, Vol. 27, No. 2, pp. 13-134, March, 1999.

[2] R. Lau, *Subword Lexicon Modelling for Speech Recognition*, Ph.D. Thesis, MIT, May 1998.

[3] R. Lau and S. Seneff, "Providing Sublexical Constraints for Word Spotting within the ANGIE Framework," *Proc. Eurospeech '97*, pp. 263—266, Rhodes, Greece, September 1997.

[4] R. Lau and S. Seneff, "A Unified Framework for Sublexical and Linguistic Modelling Supporting Flexible Vocabulary Speech Understanding," *Proc. ICSLP '98*, pp. 2443-2446, Sydney, Australia, December, 1998.

[5] S. Seneff R. Lau, and H. Meng, "ANGIE: A New Framework for Speech Analysis based on Morpho-phonological Modelling," *Proc. ICSLP '96*, pp. 110-113, Rhodes, Greece, 1996.

THESIS
RESEARCH

Heterogeneous Lexical Units for Automatic Speech Recognition

Issam Bazzi

Typical automatic speech recognition systems can recognize only words that are part of a predefined fixed size vocabulary. This constraint degrades the recognition performance when an Out-Of-Vocabulary (OOV) word is encountered since such a word will always be misrecognized. Increasing the vocabulary size does not solve the problem, since no matter how large a vocabulary is, there will always be some OOV words; names are one good example. Furthermore, increasing the vocabulary size slows down the recognition due to the larger search space. A closely related problem is recognizing partial words in speech. Partial words are usually not in the vocabulary and cannot be recognized correctly.

In this work, we explore the use of a vocabulary of subwords for recognition, such as syllables. Breaking words down into syllables reduces the vocabulary size and ameliorates the OOV problem. Figure 11 shows the word level vocabulary coverage versus the number of syllables. As we can see from the graph, a vocabulary of 1,000 syllables covers around 45,000 words, a fairly large coverage for a relatively small vocabulary.

In order to perform syllable-level recognition, we derive the lexical units from syllables instead of words, where a syllable is pronounced as the sequence of phonemes it is made of. To build the syllable bigram or trigram language model, we start with a word-level training set, and we break the words down into syllables to obtain syllable sequences for training. The use of finite state transducers (FST) in SUMMIT allows us to further integrate the syllable level

Figure 11. Word coverage versus syllable vocabulary size.

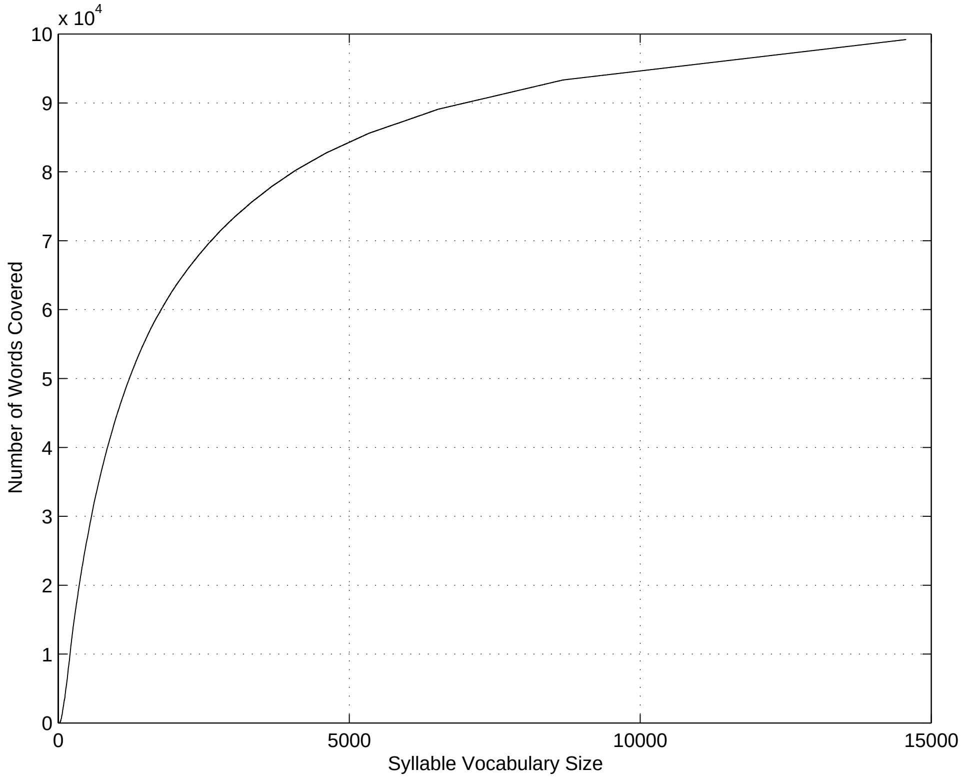


Table 2. Word error rates for word- and syllable-based recognizers.

Condition	WER
Baseline, word-level	10.4%
Syllable-level recognition with word-level composition	11.7%
Syllable graphs	13.2%

recognition with a word-level lexicon and language model by composing the output from the syllable recognizer with the word lexicon and language model [1]. The recognition is viewed as finding the best path through the composition:

A o L(syl) o G(syl) o L(wrd) o G(wrd)

where A represents the acoustic component, L represents the lexicon and G the grammar.

We ran preliminary experiments on the JUPITER weather domain. We started with a word lexicon of 1956 words. The resulting syllable vocabulary was 1624 syllables. We used the same acoustic models from the word recognizer [1] as well as the same pronunciation model. We trained the language model from 16,601 utterances of syllables. Table 2 shows some preliminary results of our preliminary tests on a JUPITER test set of 1167 utterances.

When we perform recognition with a word-level recognizer, the baseline word error rate (WER) is 10.4%. Using the syllable recognizer with a syllable trigram language model, and applying the composition with a word level lexicon and language model, the WER is 11.7%. we also experimented with a different approach where recognition was performed in two stages. In the first stage, we use a syllable-level recognizer to generate a syllablegraph, then in the second stage, we compose this graph with a word lexicon and language model. The WER we obtained using syllable graphs is 13.2%.

Considering the fact that the syllable-based framework is less constrained than the word-based framework, we believe that these preliminary results are quite encouraging. For our future work, we are looking into building a domain-independent syllable recognizer of a larger inventory of syllables. This recognizer can then be combined with a domain specific word level lexicon and language model for better performance. We are also looking into combining different lexical units within the same recognizer, mainly words and syllables. This way we can include the most frequent words and syllables in the vocabulary while allowing for the recognition of new words as well as the recognition of partial words. We are currently developing a vocabulary selection method for merging words and syllables within the same lexicon.

Reference

[1] J. Glass, T.J. Hazen, and I.L. Hetherington, "Real-Time Telephone-Based Speech Recognition in the JUPITER Domain," *Proc. ICASSP '99*, pp. 61-64, Phoenix, AZ, March 1999.

Towards Multi-Domain Speech Understanding Using a Two-Stage Recognizer

Grace Chung

Spoken language systems today are increasingly employed as human-computer interfaces for information access and interactive problem solving tasks. These technologies are accomplished through integrating speech recognition and language understanding components, and challenging research problems have arisen from contemplating the design of their architectures. For one, conversational systems that retrieve information from a broad variety of sources and services, such as on-line databases, must operate across multiple domains and allow users to switch seamlessly and transparently among them. Secondly, their ability to handle unknown words becomes very important. The experience of other researchers has shown that it is inevitable for systems to encounter previously unseen words [3], despite very large vocabularies. This is even more relevant in real-world applications, where it is not possible to anticipate all words used by all potential users, and information being accessed can change frequently and unpredictably.

One approach is an architecture that supports a flexible and easily extensible vocabulary. With conventional systems, developed over single domains, vocabularies tend to be closed or fixed. When unknown words emerge at the input, these systems can only propose errorful hypotheses or reject the utterance altogether. A more useful system would detect new words, deduce automatically their acoustic, phonological and linguistic properties, and subsequently acquire them into the lexicon.

Our work proposes an approach that is intended to realize this vision. Consider a two-stage architecture. The initial stage consists of a domain-independent core recognition engine which only utilizes acoustic and general linguistic knowledge to produce hypotheses. Trained on several large corpora, this recognizer codifies general English morphological and syllabic information. It is our belief that within the linguistic hierarchy, information up to and including the syllable level can be valuable in enforcing constraints in recognition while maintaining generality without confining the system to any fixed vocabulary items. By training on a large number of syllables, in an attempt to maximize coverage, and thereby accumulating general knowledge of English sublexical structures, this recognizer serves as a first pass whose function is to prune away a large portion of the search space. Moreover, it is capable of providing probabilistic support for novel words (that are consistent with English word structure), false starts and partial words. A recognizer with only phonemic level information would not constrain the space sufficiently, giving rise to a large number of hypotheses. On the other hand, using word level units soon becomes unwieldy because no word level recognizer can cover all novel constructions and partial word possibilities. The output of the front-end is a subword lattice which is then processed by a suite of domain-dependent speech understanding modules. With a reduced search space and thereby more manageable computational requirements, each of these modules utilizes higher level linguistic information such as domain-specific natural language (NL) models which account for dialog context. The final decision for the best meaning representation is mediated by a top-level decision algorithm.

In [1], we introduced a preliminary system that experimented with the intelligent integration of structured linguistic knowledge into a recognizer. We developed a two-stage system which applied syllable-level constraints within a first pass, and word-level and natural language (NL) constraints in the second pass. The first-pass system produced an N-best list, from which a phonetic lattice was subsequently generated. In [2], we have subsequently designed and implemented an improved system in which the first stage utilizes subword level linguistic knowledge and outputs a high-quality phonetic lattice directly, obviating the need for an intermediate N-best stage.

To generate the phonetic lattice, our system design folds hierarchical linguistic knowledge into a finite-state transducer (FST) representation. As in [1], the first pass is a syllable-level speech recognizer. A major step in performance improvements has been achieved by incorporating the probability models derived from ANGIE, our trainable hierarchical sublexical framework. We have transformed ANGIE's representation into an FST framework which can be easily integrated with our most recent FST-based version of SUMMIT via a composition algorithm. The ANGIE FST with phone input labels is precomposed with a syllable trigram FST, enabling fast computation at run-time.

These new developments allow the first pass to model phonological processes probabilistically while retaining computational tractability. This is because the dynamic parsing of ANGIE has been replaced by a flattened FST representation with ANGIE-derived probabilities precomputed on the arcs from training data. This FST representation preserves the benefits of sharing probability space, a characteristic inherent within ANGIE.

In comparison with the lattices in [1], the current system is fundamentally different. Our phonetic lattices have no knowledge of time and are optimized by the tools available for FSTs. As in [1], our second-pass search tightly couples ANGIE constraints and TINA -based NL contraints via the stack decoder search algorithm. Here our phonetic lattices have been topologically reordered, and the search proceeds along these ordered nodes.

Our work is conducted in the JUPITER domain (world-wide weather information.) On a development set, the first pass of our current system gives a substantial improvement in syllable recognition compared with the results reported in [1] from 12.8% syllable error rate down to 9.3% syllable error rate, a relative error reduction of 27.3%.

Our complete baseline system consists of a SUMMIT recognizer, which outputs an N-best list, and an understanding component, which subsequently selects the best hypothesis from this list based on NL constraints provided by the TINA framework. For evaluation, we use an understanding error rate described in [1]. The baseline system achieves an understanding error rate of 20.8% on a new test set. Our new system can be evaluated with and without TINA NL constraints in the second pass. Without TINA constraints, it achieves a 16.9% understanding error rate, which improves to 15.9% with TINA integrated. This is a 23.6% relative improvement over the baseline system.

References

[1] G. Chung and S. Seneff, "Improvements in Speech Understanding Accuracy through the Integration of Hierarchical Linguistic, Prosodic, and Phonological Constraints in the JUPITER Domain," *Proc. ICSLP '98*, Sydney, Australia, Dec. 1998.

[2] G. Chung and S. Seneff, "Towards Multi-Domain Speech Understanding Using a Two-Stage Recognizer," *Proc. Eurospeech '99*, Budapest, Hungary, Sept. 1999.

[3] I.L. Hetherington. *The Problem of New, Out-of-Vocabulary Words in Spoken Language Systems.* Ph.D. Thesis, MIT Department of Engineering and Computer Science, October 1994.

Hetergeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition

Andrew Halberstadt

The acoustic-phonetic modeling component of most current speech recognition systems calculates a small set of homogeneous frame-based measurements at a single, fixed time-frequency resolution. This thesis presents evidence indicating that recognition performance can be significantly improved through a contrasting approach using more detailed and more diverse acoustic measurements, which we refer to as heterogeneous measurements.

This investigation has three principal goals. The first goal is to develop heterogeneous acoustic measurements to increase the amount of acoustic-phonetic information extracted from the speech signal. Diverse measurements are obtained by varying the time-frequency resolution, the spectral representation, the choice of temporal basis vectors, and other aspects of the preprocessing of the speech waveform. The second goal is to develop classifier systems for successfully utilizing high-dimensional heterogeneous acoustic measurement spaces. This is accomplished through hierarchical and committee-based techniques for combining multiple classifiers. The third goal is to increase understanding of the weaknesses of current automatic phonetic classification systems. This is accomplished through perceptual experiments on stop consonants which facilitate comparisons between humans and machines.

Figure 12 shows the error rate reduction obtained in a TIMIT phonetic classification task when combining all possible subsets of eight classifiers using an independence assumption. Error rates are shown as a function of the number of classifiers in the

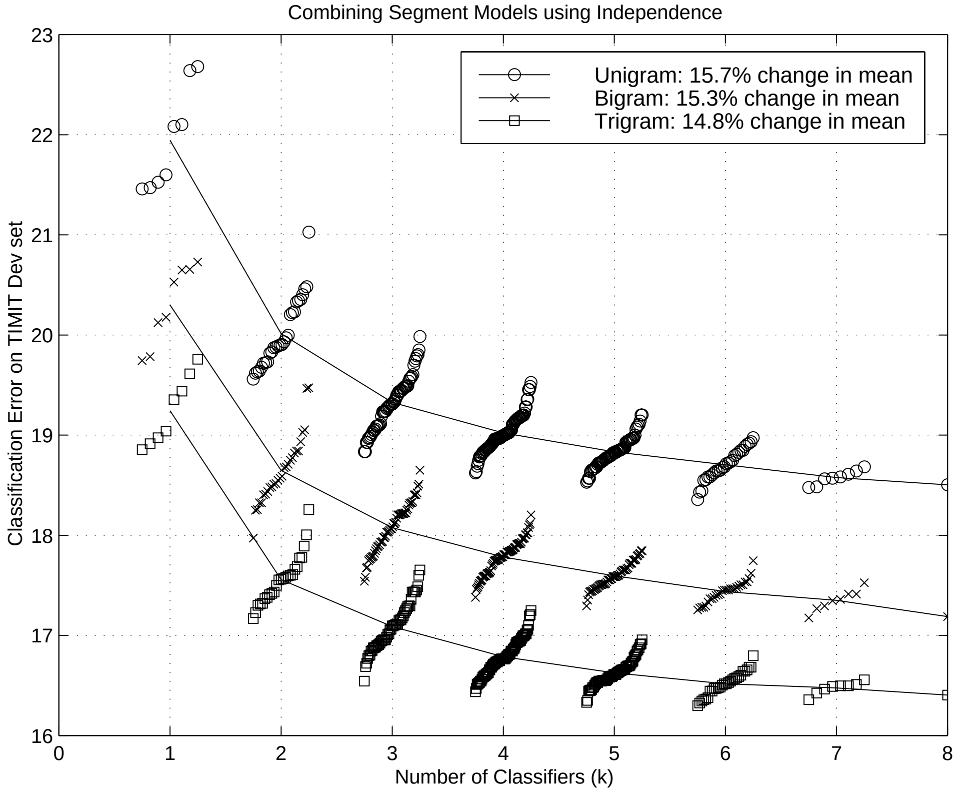


Figure 12. Performance of classifier combination using an independence assumption for all possible subsets of 8 classifiers. Error rate is shown as a function of the number of classifiers in the set. Results are shown for unigram, bigram, and trigram phonotactic model conditions.

subset. In the figure, individual data points are evenly spaced along the x-axis in the vicinity of the appropriate value of k, where k represents the number of classifiers being combined. Lines connect the mean values. Unigram, bigram, and trigram phonetic phonotactic models were used. The total number of experiments for each phonotactic model is 255. As k varies from 1 to 8, the number of individual classification experiments is 8, 28, 56, 70, 28, 8, 1, respectively. The results indicate that combining classifiers using independence produces favorable trends with all three phonotactic models, with relative error rate reduction of about 15% on average.

In further empirical work, systems using heterogeneous measurements and multiple classifiers were evaluated in phonetic classification, phonetic recognition, and word recognition tasks. On the TIMIT core test set, these systems achieved error rates of 18.3% and 24.4% for context-independent phonetic classification and context-dependent phonetic recognition, respectively. These results are the best that we have seen reported on these tasks. Word recognition experiments using the corpus associated with the JUPITER telephone-based weather information system showed 10-16% word error rate reduction, thus demonstrating that these techniques generalize to word recognition in a telephone-bandwidth acoustic environment.

This thesis fits into the growing literature on pattern recognition using an ensemble of classifiers [4]. The acoustic-phonetic motivation for employing ensemble classification for acoustic modeling in speech recognition is developed through comparisons between human and machines in phonetic classification tasks, and through signal processing knowledge of the fundamental tradeoffs in time-frequency resolution. In addition to the experiments reported here, the use of multiple acoustic information sources and multiple classifiers has led to encouraging results in a wide variety of speech recognition experiments reported in the literature [5, 6, 7]. Further research will explore new ways to combine classifier ensembles and search for corresponding high-dimensional acoustic feature spaces which are made practical through ensemble methods.

References

[1] A. K. Halberstadt and J. R. Glass, "Heterogeneous Measurements for Phonetic Classification," *Proc. Eurospeech* '97, pp. 401—404, Rhodes, Greece, September 1997.

[2] A. K. Halberstadt and J. R. Glass, "Heterogeneous Measurements and Multiple Classifiers for Phonetic Classification," *Proc. ICSLP '98*, Sydney, Australia, December 1998.

[3] A. K. Halberstadt. *Heterogeneous Measurements and Multiple Classifiers for Phonetic Classification*. Ph.D. Thesis, MIT Department of Electrical Engineering and Computer Science, November 1998.

[4] D.J. Miller and L. Yan, "Critic-Driven Ensemble Classification," *IEEE Trans on Signal Processing*, Vol. 47, No. 10, pp. 2833-2844, October 1999.

[5] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding 1997*, (http://www.nist.gov/speech/papers.htm)

[6] NIST, "NIST September 1998 Hub5-E Benchmark Test Results for Recognition of Conversational Speech over the Telephone, in English," (Available via ftp from jaguar.ncsl.nist.gov, in /evaluations/hub5/sept98)

[7] K. Kirchhoff and J. Bilmes, "Dynamic Classifier Combination in Hybrid Speech Recognition Systems using Utterance-level Confidence Values," *Proc. ICASSP '99*, pp. 693-696, Pheonix, AZ, March 1999.

Word and Phone Level Acoustic Confidence Scoring

Simo Kamppari

While the recognition performance of speech recognition systems improves constantly, the systems remain far from perfect. Accepting the fact that mistakes in recognition will be made, it is desirable to determine when these mistakes occur. The purpose of word level confidence scores is to indicate a certainty that a given word is correctly recognized. Access to robust word level confidence scores enables improvements in various components of existing spoken language systems and may also lead to entirely new functionalities. This research has focused on developing acoustic based phone and word level confidence scores within a segmentation based recognizer SUMMIT [2,4], and exploring possible applications for these scores. The work has been implemented on the JUPITER [3] weather information system which provides weather information regarding approximately 500 cities around the world.

The goal of word level confidence scoring is to attach a confidence value with each word in the recognition hypothesis, indicating the likelihood that the word is correctly recognized. An ideal confidence metric would be a simple binary flag; it would always correctly point out which words in a hypothesis are correct and which are wrong. As achieving such performance is impossible in the absence of perfect speech recognition, a probabilistic measure is a more feasible goal. The probabilistic measure is non-binary and reflects the likelihood that a word is correctly recognized.

This research approaches the word level confidence scoring problem as a derivative of the phone level confidence scores. The phone level confidence scores are calculated from the acoustic observations associated with each phone. The basic phone level confidence feature is a measure of how well a proposed boundary model accounts for an acoustic observation relative to a generic *catch-all* model. The *catch-all* model represents all the possible boundary models in JUPITER, and it is created by simply aggregating all the individual JUPITER boundary models. This feature takes on the form of a likelihood ratio $C_{ll}(c_i|\vec{x})$ where c_i is the proposed model and $\vec{x}$ is the acoustic observation. It can alternatively be viewed as a ratio of two scores, the score of the acoustic observation given the proposed model and the score of the acoustic observation given the *catch-all* model, as shown in Equation 1 . In addition to this feature, a few other phone level features are proposed.

$$C_{ll}(c_i|\vec{x}) = \frac{p(\vec{x}|c_i)}{p(\vec{x})} = \frac{p(\vec{x}|c_i)}{\sum_{j=1}^{N_c} p(\vec{x}|c_j)P(c_j)} \tag{1}$$

The word level confidence scores are derived from combining phone level confidence scores in various ways. The most basic method for generating a word level confidence measure is an arithmetic mean of the phone level scores within a word

$$C_{am}(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k, c_1, c_2, \ldots, c_k) = \frac{1}{N_L} \sum_{k=1}^{N_L} C_p(c_k|\vec{x}_k)$$

Other methods, like geometric means and various *min/max* functions, were also used to derive word level confidence measures. Additional word level features can also be used to augment these phone derived confidence scores. All together 12 different features were proposed as measures of word level confidence.

To derive meaningful information from the set of 12 unique features, Fisher Linear

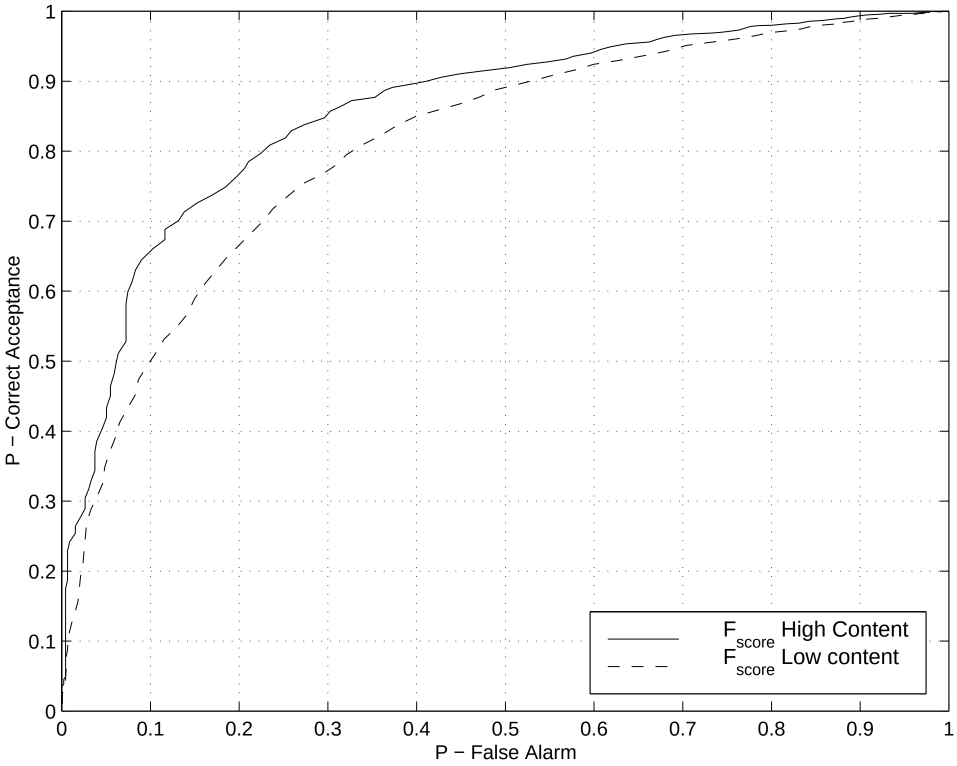


Figure 13. Relative performance of word class High and Low.

Discriminant Analysis [1] was used to project the 12 dimensional feature space into a single maximally discriminating dimension. The theoretical performance of the single dimension is capped at the performance of the 12 dimensional space; however this limit in performance is outweighed by the simplification of the analysis.

The performance of the word level confidence scores was explored in terms of the information content of words. The results were encouraging as confidence scores of words with higher information content performed significantly better that the scores of low information content words. In terms of actual applications the performance on content words is much more important than that of non-content words. Ultimately, most systems are interested in correctly *understanding* a user rather than correctly recognizing them. This translates to a high tolerance in misrecognizing words which do not affect the meaning of an utterance, and a low tolerance for miserecognizing words with meaning. Figure 13 illustrates the differences in performance for high and low content words in a form of a Receiver Operating Characteristic (ROC), where F_{score} is the confidence metric derived from the 12 word level features using FLDA.

The above results, along with the fact that the computation of the confidence scores was made computationally reasonable by using compact approximates of the *catch-all* model, makes the confidence scores viable for use in practical applications. This research has worked in incorporating the confidence scores into the parsing and understanding component in TINA. The goal is to replace a current *ad hoc* method of word graph node scoring with the confidence scores. The confidence scores have been incorporated in various ways, and the performance has varied greatly. No significant gains over the *ad hoc* method were realized; however further work may lead to more efficient ways for using the scores. Further research is necessary to fully assess the value of the confidence metrics in this context.

In addition to this application, many others exist. User feedback has great potential for confidence scores. By utilizing robust confidence scores, the human machine interaction can be made both more natural and robust. For example, the system may potentially prompt the user to repeat or clarify parts of an utterance the computer may have not *heard* well. So far this research has provided much insight into confidence scoring analysis and has formed a solid foundation from which further research into various applications can take place.

References

[1] R. Duda, and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1973.

[2] J. Glass, "A Probabilistic Framework for Feature-Based Speech Recognition," *Proc. ICSLP '96*, Philadelphia, PA, 1996.

[3] J. Glass, T. Hazen and L. Hetherington, "Real-Time Telephone-Based Speech Recognition in the JUPITER Domain," *Proc. ICASSP '99*, pp.61-64, Pheonix, AZ, March 1999.

[4] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, "Recent Progress on the SUMMIT System," *Proc. DARPA Speech and Natural Language Workshop*, Hidden Valley, PA, 1990.

The seMole: A Robust Framework for Accessing Information from the World Wide Web

Hyung-Jin Kim

Although many people have labeled the World Wide Web as the largest database ever created, very few applications have been able to use the web as a database. This is because the web is dynamic: web pages change constantly, sometimes on a daily basis. I propose a system called the "seMole" that aims to solve this problem by providing a semantic interface into the web. The semantic interface uses the semantic content on web pages to map very high-level concepts, such as "weather reports for Boston" to low-level requests for data (such as getting the text in the third 'A' tag in a web page). Therefore, even though web pages change, the seMole will still be able to find information on them.

The seMole will robustly access a web page by taking advantage of the topology of its underlying HTML. When web pages get updated, the information that is presented usually retains the same structure. For example, when the CNN Weather Data site changed in November of 1997, its facade changed, but it still continued to present the same information, namely data about the current conditions of a city and a four-day forecast. Furthermore, although the HTML structure of this new page was drastically different, the weather information was still grouped in the same way (i.e. high and low temperatures were still presented next to each other).

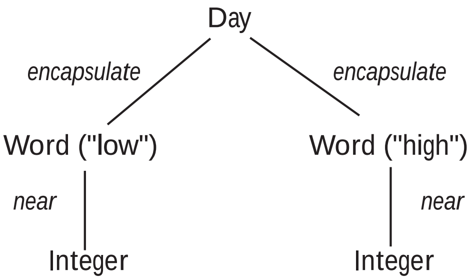


Figure 14. Semantic template for CNN weather.

The seMole uses semantic templates to access information from web pages. In the weather example, to gather all of the 4-day forecasts of a city, the template in Figure 14 is used. The seMole takes this template and matches it to the data on the web page. This template essentially drills down through high-level concepts presented on the web page. First, it finds a "day" word e.g., "Monday" on the web page and then it tries to find the words "low" and "high" that are associated with that word. Finally, it finds the integers that are most closely located to the words "low" and "high". Since this semantic template is abstracting away from the HTML structure, this template would have found the same temperature information before and after the change (Figure 15). Notice that this template follows what a human does to gather the same information: first, he searches for a specific day and then he searches for the temperatures besides the words "high" and "low."

In order to make use of semantic templates, the seMole will require the following facilities: a taxonomy of data descriptors and a library of relationship descriptions. A taxonomy of data descriptors is used to describe all possible data or recognizable features on a web page. In our weather template, we used the names "integer" and "day" to describe the data we are looking for. In order for the seMole to access many different types of web pages, a large library of data types needs to be created. One can imagine extending this taxonomy to incorporate concepts of "state," "country," and "car_name". This taxonomy can be hierarchical in that a semantic idea can be built on top of other semantic ideas, making them highly scalable and re-usable. A library of relationship descriptors describes all the ways in which

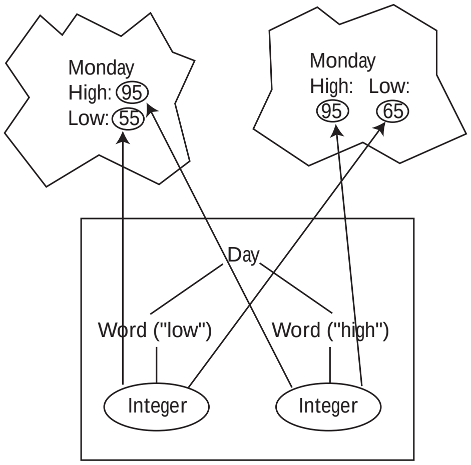


Figure 15. Mapping of the semantic template to two versions of the weather page (note: not necessarily the CNN weather page).

features of a web page can relate to each other. Descriptors such as "near" and "on_top_of" are simple examples of relationship descriptors. More complicated descriptors include "encapsulate" which not only define how one datum is positioned relative to another, but also how the fonts of each datum are related to each other (words with large, bolded fonts encapsulate smaller fonted words following them).

The seMole is potentially a very robust and simple interface for applications to access the web. By "lifting" semantic concepts found on a web page away from the HTML structure, the seMole will be able to gather information from web pages even when these pages change. In many ways, semantic templates attempt to mimic what a human does to find information. By using concepts instead of HTML tags to find information, the seMole is using web pages as they were meant to be used: by the human eye.

References

[1] H.-J. Kim. *Semole: A Robust Framework for Gathering Information from the World Wide Web*. M.Eng. thesis. MIT Department of Electrical Engineering and Computer Science, September 1998.

[2] H.-J. Kim, and L. Hetherington, "SeMole: A Robust Framework for Gathering Information from the World Wide Web," *Proc. ICSLP '98*, Sydney, Australia, December 1998.

Analysis and Modeling of Non-Native Speech for Automatic Speech Recognition

Karen Livescu

The performance of automatic speech recognizers has been observed to be dramatically worse for speakers with non-native accents than for native speakers. This poses a problem for many speech recognition systems, which need to handle both native and non-native speech. The problem is further complicated by the large number of non-native accents, which makes modeling separate accents difficult, as well as the small amount of non-native speech that is often available for training. Previous work has attempted to address this issue by building accent-specific acoustic and pronunciation models or by adapting acoustic models to a particular non-native speaker.

In this study, we have examined the problem of non-native speech in a speaker-independent, large-vocabulary, spontaneous speech recognition system for American English, in which a large amount of native training data and a relatively small amount of non-native data are available. We have investigated some of the major differences between native and non-native speech and have attempted to modify the recognizer to better model the characteristics of non-native data. This work has been performed using the SUMMIT speech recognition system [1] in the JUPITER weather information domain [2].

We first examine the modification of acoustic models for better recognition of non-native speech. In order to take advantage of the large amount of native training speech available to us, as well as the information that we can gather from the small amount of non-native speech, we use a model interpolation approach. In acoustic model interpolation, a single model is built by taking a weighted average of the probability distribution functions corresponding to several models. In this case, we interpolate two models: One which has been trained on native data, and is therefore robust but not matched to the testing data, and another which has been trained on the non-native data, and is therefore less robust but better matched. We have compared interpolated models with varying component weights to a set of baseline models trained on a pooled set of all of the native and non-native training data. We have found that we can obtain a significant gain in recognition accuracy for a large range of model weights. The lowest word error rate we have obtained using this method on a non-native test set is 19.2%, a relative reduction of 8.1% from the baseline error rate of 20.9%. This minimum occurs for a non-native weight of 0.54 and a native weight of 0.46. However, the location of this minimum is likely to depend on the relative amounts of native and non-native training data (in our case, the native utterances outnumbered the non-native ones by a factor of about 12).

In the area of lexical modeling, we have performed a small study of native and non-native word pronunciation using manual phonetic transcriptions. We have made detailed hand transcriptions of about 200 utterances, equally divided between native and non-native speakers. From both subjective inspection of these transcriptions and automatic alignments between the transcriptions and the JUPITER lexicon, we have observed some systematic differences between natives and non-natives. This suggests that we may be able to model non-native pronunciation patterns by incorporating additional rules into the recognizer's lexicon. However, discovering such rules and applying them to the lexicon is laborintensive and prone to human error. We have therefore attempted to model pronunciation phenomena using an approach in which patterns are automatically learned from the non-native training data.

In this initial investigation, we constrain the patterns to context-independent substitutions, insertions, and deletions of phones, independent of context. The probabilities of these phonetic confusions are estimated from non-native training data by aligning automatically-generated phonetic transcriptions with the baseline lexicon. The confusions are incorporated into the recognizer using a simple weighted finite-state transducer encoding the probability of each confusion. We have experimented with various training mechanisms within this general approach, and have obtained relative reductions in error rate of up to 10.0% over the baseline recognizer on the non-native test set. Using both phonetic confusions and interpolated acoustic models, we have further reduced the word error rate to 12.4% below baseline. These results were obtained using a very simple model with only context-independent phonetic confusions. We believe that this approach has many useful applications and extensions, including context-dependent modeling and data-driven learning of more complex phonological rules. These methods can be applied more widely, not only to non-native pronunciation patterns but to lexical and phonological modeling in general.

Finally, we have performed a study of language model differences between native and non-native speakers in the JUPITER domain. We have compared the entropies of native and non-native utterances with respect to n-gram language models and studied the correlation between entropy and word error rate. We have found that, within the resolution of our analysis, language model differences do not account for a significant part of the degradation in recognition performance between native and non-native test speakers. However, a more detailed analysis may reveal more specific differences and suggest possible modifications to the language model. As a starting point, it would be interesting to attempt to interpolate separate language models trained on native and non-native speakers.

This work has shown that it is possible to improve recognition accuracy on non-native speech while treating all non-native speakers as a single group. This is potentially useful for the development of speech recognition systems in domains in which non-native data are limited, or in which it is undesirable for other reasons, such as computational or time considerations, to separate non-native speakers into specific accents. Although the results show significant gains in performance when modeling all non-natives as a group, there are accent- and speaker-specific phenomena that could be better modeled. A system that uses a single set of baseline models for all non-native speakers and adapts to the characteristics of a particular test speaker may be a good way to address the complexities of non-native speech. For a conversational system like JUPITER, in which the speaker usually remains constant throughout a multi-utterance interaction, an incremental adaptation approach may be particularly suitable.

References

[1] J. Glass, J. Chang, and M. McCandless, A Probabilistic Framework for Feature-based Speech Recognition. *Proc. ICSLP '96*, pp. 2277-2280, Philadelphia, PA, October 1996.

[2] J. R. Glass and T. J. Hazen, Telephone-based Conversational Speech Recognition in the JUPITER Domain. *Proc. ICSLP '98*, pp. 1327-1330, Sydney, Australia, December 1998.

Toward a General Framework for Robust Understanding

Kristine Ma

The ultimate goal of the speech understanding research field is to design a speech understanding system that can degrade gracefully under all levels of adverse environment. This includes the ability to handle speech recognition error, misunderstanding error, and out-of-domain queries. As a small step towards this goal, we study the robustness of an understanding system in the in the presence of recognition error.

The Spoken Language Systems Group at MIT has traditionally used a statistically based parser called TINA [1] for spoken language understanding. The underlying approach of TINA is to first capture the full meaning representation of the sentence, then, having understood the sentence, extract the relevant pieces of information (such as city names) from the full semantic representation. Such an approach is powerful, yielding very low false acceptance rate, however at the expense of development time and robustness.

In this work, we took an approach from the other extreme, an investigation into the feasibility of using a phrase spotting system for speech understanding within a narrow application domain. Such a system, though it differs in implementation, is similar in spirit to [3]. The goal is first to compare, quantitatively, the performance of a key-phrase spotting style system versus a more syntactically constrained understanding system; second, to examine the strengths and weaknesses of each of these approaches. The eventual goal is to find a unified framework where one can leverage the strengths of both methods.

The baseline system for this project was built using the Finite State Transducer recognizer as the front-end, TINA as the back-end understanding component, and JUPITER [2], a weather information system, as the task domain.

To build the key phrase spotter, we generated 1638 sets of KEY:value pairs — 1026 geographical location names, 380 weather events, 193 dates or times, and 39 greeting or help requests. They are derived based on observing the KEY:value tags of 1021 sentences generated by TINA. This set of "words to KEY:value" mappings is then compiled into a single finite state transducer, which in turn transforms the word-graph from the recognizer's output hypothesis into concept graphs (a graph consists of KEY:value pairs such as CITY:Boston, WEATHER:snow, DATE:Saturday, etc.). The best path through this concept graph is then selected based on a combination of the sentence likelihood score from the recognizer and a bigram score on the KEYs. This language model predicts the likelihood of observing KEY i following KEY j for all i,j in the KEY set. The model is trained using 26,000 JUPITER training sentences that have been converted into strings of KEYs.

The two final systems have comparable performance in terms of concept detection error rate. Table 3 shows the evaluation results of the two systems on unseen data based on KEY:value detection. We divided the test set into 4 subsets based on whether the true transcription can be fully parsed via word-graph by TINA, robustly parsed via word-graph by TINA, keyword spotted via N-best by TINA, or cannot be parsed at all by TINA. Comparing the performance of TINA versus the phrase spotter on each subset, we observe that the phrase spotter is doing better than TINA on all counts except on sentences whose true transcription can be fully parsed. Comparing the concept error

	Recognition Error Rate			TINA parsing word-graph				Phrase spotting word-graph			
	#Sent	WER	SER	CER	DET	FA	UER	CER	DET	FA	UER
FULL_PARSE	1803	14.9%	36.0%	16.20	90.67	10.02	19.2%	17.28	87.98	8.15	22.8%
ROBUST_PARSE	220	40.5%	88.6%	42.97	75.41	24.32	57.1%	42.16	75.95	21.89	62.7%
KEYWORD_PARSE	110	80.5%	98.2%	52.55	64.96	21.17	56.3%	40.88	80.66	29.20	51.6%
NO_PARSE	105	51.7%	88.6%	100.00	0.00	1.45	100.0%	69.57	57.97	36.23	76.2%
Total	2238	26.0%	46.7%	21.66	86.60	11.69	25.3%	21.45	86.03	10.73	28.2%

Table 3. Performance comparison sof two systems in the JUPITER *domain.*

rate (CER) versus the understanding error rate (UER), we see that while the spotter is better in spotting individual concepts, TINA is better in capturing the complete meaning of a query.

From intuition, we know that both TINA and the phrase spotter have their own weaknesses and strengths. From data analysis, the contrast is even more apparent. Relying on detecting just the presence of key-phrases, a spotter will have problem rejecting out of domain sentences that contain keywords:

```
 what will the water temperature be in new york city tomorrow
  what kind of clouds will be in san jose
```

Unless the spotter has the a priori knowledge that "water temperature" and "clouds" are consider to be out of domain queries, it will misinterpret the two questions for "temperature in new york city tomorrow" and " weather in san jose", respectively.

TINA, on the other hand, has a tendency of falsely rejecting unexpectedly phrased queries that arise due to recognition errors, hesitation, restarts, the conjunction of multiple short phrases, etc. Here are some examples:

```
  what is the weather like in pittsburgh thi- this sunday
  fort lauderdale should i spell it
  good bye and have a good day bye bye
  what is up with boston
```

Using both semantic and syntactic constraints, TINA is capable of capturing not only long-distance context but also complex sentence structures. Therefore one can argue that given multiple alternatives, TINA would select a grammatically more plausible hypothesis, whereas the phrase spotter might not. However from this investigation, we found that using the KEY bigram together with the recognition likelihood scores, the spotter is capable of selecting as good a hypothesis as TINA from word graphs that contain an average size of 80 N-best. However, JUPITER could be too simple an application domain (in terms of linguistic and dialogue complexity) to demonstrate the advantage of using a full parser. The main advantages of a phrase spotter is that it is robust to real speech.

The KEY:value tags generated for the phrase spotter reported here were derived manually. However, this is in essence a named-entity tagging problem and hence can in theory be done automatically (with tagged training data) using approaches such as the Brill transformation tagger or the BBN Nymble system.

In conclusion, this study gives us a preview of the robust issue in the task of spontaneous speech understanding. Future work includes moving onto a more linguistically complex application domain, investigating the issue of accurate out of domain query rejection, and starting to look into a unified framework that can bridge the gap between these two extreme approaches with robustness as the priority issue.

References

[1] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, pp. 61-86, 1992.

[2] V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, P. Schmid, "From Interface to Content: Translingual Access and Delivery of On-Line Information," *Proc. Eurospeech '97*, pp. 2227-2230, Rhodes, Greece, 1997.

[3] W. Ward, S. Issar, "The CMU ATIS system," *Proc. of the ARPA Workshop on Spoken Language Systems Technology Workshop*, Austin, TX, pp. 249-251, 1995.

PANDA: Telephone Based Chinese Conversational System in Weather Domain

Xiaolong Mou

The motivations for implementing a telephone based Chinese conversational system in the weather domain are to explore the multi-language ability of the SLS GALAXY-II architecture, build a conversational system prototype for further speech recognition and understanding research in Chinese, and study the Chinese language specific issues in both acoustic modeling and natural language understanding.

GALAXY-II [1] is a client-server architecture developed at the MIT SLS group for accessing on-line information. Various human language processing servers are connected through a central control, the programmable hub, and mediated by the hub script. The servers are designed to be as language transparent as possible, such that implementing a conversational system in a different language would be relatively convenient [2]. In this research project, Chinese acoustic models, language models and natural language understanding rules in weather domain are developed for PANDA, the telephone based Chinese Conversational system in the weather domain.

Paraphrased speech is collected for the prototype system. In order to facilitate the data collection effort via the telephone, we have tried an automatic voice prompt generation procedure. First we define a set of weather related sentence templates with substitutable variables. For example, the template “What is the :WEATHER-DETAIL in :CITY” and the variable definition “:CITY => Lanzhou, Beijing, Shanghai,” and “:WEATHER-DETAIL => temperature, wind speed,” would allow us to generate a variety of sentences people can ask about weather. Then the system will randomly substitute the variables, and concatenate the pre-recorded voice pieces to generate the prompts over the telephone for subjects to paraphrase. This procedure turns out to be convenient and successful. We have collected about 240 minutes of speech by 20 speakers for training, and about 60 minutes of speech by 5 speakers for system development. The baseline system is built on the training data, then evaluated and improved on the development data.

MIT SLS’s SUMMIT system is used for speech recognition. SUMMIT is a segment-based speech recognizer. A pronunciation network is created from a baseform file and a set of phonetic rules. The baseform file specifies the basic pronunciation of each word in the vocabulary, and the set of phonetic rules will take actual pronunciation variations into account. The recognizer then searches the best path through the network, with the guidance of acoustic models and language models. In this project, efforts have been made to improve the language model by trying different smoothing techniques and combining word models with class models.

Basically, smoothing is a technique to adjust the maximum likelihood estimation when the training data are sparse, such that more accurate probability estimation can be achieved. For *n*-gram models, a vocabulary of size V will have V^n *n*-grams; thus the sparse data problem is one of the major concerns when applying *n*-gram language modeling technique. Smoothing generally will adjust low probabilities upward and high probabilities downward. It attempts to improve the accuracy of the language model as a whole by adjusting the model for the unseen data. Several more sophisticated smoothing methods have been tried as opposed to the currently used simple interpolation smoothing. Experiments show

Table 4. Recognition performance with bigram and trigram language models.

	WER(%)	Substitution(%)	Insertion(%)	Deletion(%)
Bigram Language Model	11.9	6.5	3.2	2.2
Trigram Language Model	8.4	4.3	2.1	2

that Katz smoothing performs much better when the training corpus is large.

In addition, we have tested a novel back off idea. Instead of either using a word n-gram model or a class *n*-gram model, we can combine them. When a word has a high count, the maximum likelihood estimation is reliable, and then it is used directly; for those words with a small number of occurrences, we use a class *n*-gram model as the back off model. The idea is to keep the relatively accurate high probability word estimation, and smooth the low probabilities using a class *n*-gram model, thus combining the advantages of both models. The results show that the hybrid language model achieves lower perplexity and word error rate than either of the original models does.

We use TINA for natural language understanding. TINA is a robust probabilistic natural language system based on a context-free grammar augmented with a set of features to enforce syntactic and semantic constraints. Chinese-specific rules are developed to handle Chinese weather related queries. It is particularly important to use TINA's trace mechanism to deal with movement phenomena in Chinese queries. As an example, consider the following query: Lanzhou you3 xue3 ma5? which means Is there any snow in Lanzhou? The word Lanzhou is moved to the beginning of the sentence. With TINA'S tracing capability, it can correctly generate a semantic frame with an "in Lanzhou" predicate.

One unique problem of Chinese speech processing arises from the fact that the language does not have explicit word boundaries. Without the natural language understanding unit, the speech recognizer would have to solely rely on the language model to pick up one promising word segmentation. However, the typical n-gram language model does not have the ability to model long distance constraints; thus word segmentation errors are quite frequent in this case. We can use TINA here to further detect and correct such errors in the context of the whole sentence, a much more robust strategy.

Table 4 shows the preliminary recognizer evaluation results. A 20K word corpus of weather related articles and conversations is used to train the language model.

Figure 16 shows the iterative training procedure. We use the trained acoustic models to refine the phonetic transcription, then re-train the models. As we can see from the figure, the iterative training effectively reduced the word error rate.

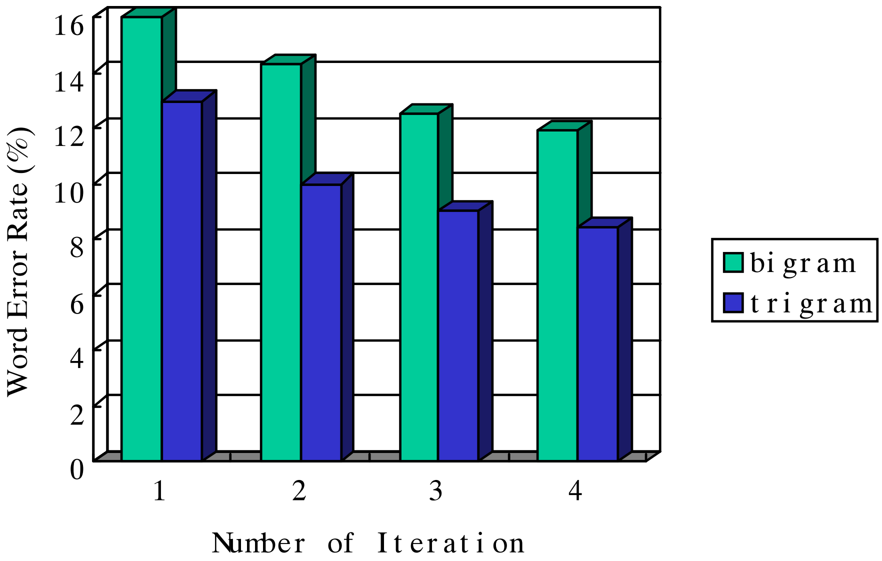


Figure 16. Automatic iterative transcribing and training.

In summary, a Chinese telephone based conversational system prototype for accessing weather information is implemented. The prototype seems to have a satisfactory performance with limited training data. More work needs to be done to improve it, including collecting more spontaneous data, developing a more realistic and comprehensive set of rules for TINA, and studying more about Chinese specific issues.

References

[1] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "GALAXY-II: A Reference Architecture for Conversational System Development, " *Proc. ICSLP '98*, Sydney, Australia, December 1998.

[2] J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue, "Multilingual Spoken-Language Understanding in the MIT VOYAGER System," *Speech Communication*, Vol. 17, No. 1, pp. 1-18, March 1995

Subword-based Approaches to Spoken Document Retrieval

Kenney Ng

Our research explores approaches to the problem of spoken document retrieval (SDR), which is the task of automatically indexing and then retrieving relevant items from a large collection of recorded speech messages in response to a user specified natural language text query. We investigate the use of subword unit representations for SDR as an alternative to words generated by either keyword spotting or continuous speech recognition. Our investigation is motivated by the observation that word-based retrieval approaches face the problem of either having to know the keywords to search for a priori, or requiring a very large recognition vocabulary in order to cover the contents of growing and diverse message collections. The use of subword units in the recognizer constrains the size of the vocabulary needed to cover the language; and the use of subword units in the indexing allows for the detection of new user-specified query terms during retrieval. Three research issues are addressed. First, what are suitable subword units and how well can they perform? Second, how can these units be extracted from the speech signal in a reliable and efficient manner? And third, how can the indexing and retrieval methods be modified to effectively deal with the fact that the speech recognition output is errorful?

We address the first issue in our Eurospeech '97 paper [2] where we explore a range of subword units of varying complexity derived from phonetic transcriptions. We find that subword units are able to capture enough information to perform effective retrieval. With the appropriate subword units it is possible to achieve performance comparable to that of text-based word units if the underlying phonetic units are recognized correctly. In our ICASSP '98 paper [3], we explore the second issue by developing a phonetic speech recognizer, running it on the spoken documents, processing the recognition output to create subword units for indexing and retrieval, and then examining the effects of recognition errors on retrieval performance. We find that in the presence of phonetic recognition errors, retrieval performance degrades but many subword units are still able to achieve reasonable performance even without the use of any error compensation techniques.

Our recent work focuses on the third issue by investigating robust indexing and retrieval methods in an effort to improve retrieval performance when there are speech recognition errors [4]. We examine a number of methods that take into account the characteristics of the recognition errors and try to compensate for them. In the first approach, the original query representation is modified to include similar or confusible terms that could match erroneously recognized speech; these terms are determined using information from the phonetic recognizer's error confusion matrix. The second approach is a generalization of the first method and involves developing a new document-query retrieval measure using approximate term matching designed to be less sensitive to speech recognition errors. In the third method, the document representation is expanded to include multiple recognition candidates (e.g., N-best) to increase the chance of capturing the correct hypothesis. The fourth method modifies the original query using automatic relevance feedback to include new terms found in the top ranked documents. The last method involves the "fusion" or combination of information from multiple subword unit representations. We study the different methods individually and then explore the effects of combining them. Experiments on radio broadcast news data show that each method can help retrieval performance and that using a combination of these methods can improve performance by over 20%. Future work in this area include investigating more sophisticated probabilistic models for approximate matching; examining non-linear methods for combining different subword units; and exploring ways to integrate recognizer likelihood and confidence scores more directly into the indexing and retrieval process.

Our other recent efforts have focused on the development of a novel probabilistic information retrieval model designed to facilitate the tighter integration of the speech recognition and information retrieval components within the context of the SDR task. The goal is to move away from the standard approach of using the speech recognizer to transform the speech into text transcriptions and then feeding those directly into a full text retrieval system. Although this approach of simply cascading the two processes is modular, it may also be sub-optimal since each component is originally designed with different objectives and assumptions in mind. The speech recognizer is designed to output the most likely word sequence corresponding to a given set of acoustic observations. The full-text retrieval engine is designed to index and search a collection of text documents and expects error-free text representations as input. Because the speech recognizer is not perfect, there is a mismatch between what the recognizer outputs and what the retrieval system expects as input. Information about the uncertainty of the recognized words is likely to be useful in the indexing and retrieval process. The question is then: what is the best way to get this type of information out of the recognizer and how can it be effectively used in the retrieval model.

One approach is to develop a retrieval model that uses information that can be directly computed by the speech recognizer. We propose a novel probabilistic information retrieval model that scores documents based on the relative change in the document likelihoods, expressed as the likelihood ratio of the conditional probability of the document given the query and the prior probability of the document before the query is specified. The idea is that documents that become more likely after the query is specified are probably more useful to the user and should score better and be ranked ahead of those documents whose likelihoods either stay the same or decrease. The document likelihoods are computed using statistical language modeling techniques which eventually make use of the probabilistic quantity $p(t|d)$, the probability that term t occurs in document d. The goal of the speech recognizer can be modified to generate these term occurrence probabilities given the acoustic observations instead of finding the most likely word sequence. In this way, the interface between the speech recognition and retrieval components are matched and the goals of the two components are consistent with the overall goal of the combined system. To benchmark the performance of the new retrieval model, we use the standard ad-hoc text retrieval tasks from the TREC-6 and TREC-7 text retrieval conferences [1]. Experimental results indicate that our model is able to achieve state-of-the-art retrieval performance on these tasks. Continuing work in this area includes modifying the speech recognizer to output the desired term occurrence probabilities given the acoustic observations, integrating the modified speech recognizer with the new retrieval model, and then evaluating the performance of the combined system.

References

[1] D. K. Harman, ed., Seventh Text Retrieval Conference (TREC-7) Gaithersburg, MD, USA, National Institute for Standards and Technology, 1998. NIST-SP 500-242.

[2] K. Ng and V. Zue, "Subword Unit Representations for Spoken Document Retrieval," *Proc. Eurospeech '97*, Rhodes, Greece, pp. 1607-1610, Sept. 1997.

[3] K. Ng and V. Zue, "Phonetic Recognition for Spoken Document Retrieval," *Proc. ICASSP '98*, Seattle, WA, USA, pp. 325-328, May 1998.

[4] K. Ng, "Towards Robust Methods for Spoken Document Retrieval," *Proc. ICSLP '98*, Sydney, Australia, Nov. 1998.

Toward the Automatic Transcription of General Audio Data

Michelle S. Spina

The last few years have been an exciting time in the "information age." We have seen an enormous growth in the amount of information available electronically to users, and as the popularity of the World Wide Web continues to grow, we will continue to see further increases. Until recently, the vast majority of this information has been text-based, from sources such as quarterly reports, text-based web pages, catalogs, theses, conference proceedings, weather reports, etc. Recently, in addition to the increase in the amount of information available to users we have also seen an increase in the type of information available. In addition to text-based data, we now are gaining access to images, video and audio data from sources such as television, movies, radio and meeting recordings. Access to these multimedia sources of information would allow us to fulfill such requests as "Play me the speech in which President Kennedy said `Ich bin ein Berliner'," "Show me the news segment in which President Bush addressed the troops in Desert Storm," or "Summarize Victor's conclusions in the last staff meeting."

These multimedia sources of information have presented us with new research challenges. Much research has been done on the problem of selecting relevant items from a large collection of text-based materials. Traditionally, the key words present in the text documents are used to index and describe the content of the documents, and information retrieval techniques have been developed to efficiently search through large collections of such data. Only recently has there been work addressing the retrieval of information from other media such as images or audio. Unlike text-based data, however, multimedia data sources do not have such a natural way to index or describe their content. Given that these multimedia sources of information are becoming a large portion of the available data to users, it is critical to the success of future information systems to have an ability to automatically index and describe the content of multimedia information.

If we concentrate on general audio data (GAD) from sources such as radio, television, movies, meeting recordings, etc., one natural solution to this problem would be to generate a full transcription of the audio, in effect converting the audio data to text data, such as is done with closed captioning. Traditional text-based information retrieval techniques could then be used directly on the transcriptions. A manual approach to this problem seems intractable, due to its tedious and time-consuming nature. However, in recent years we have made great strides in automatic speech recognition (ASR) as a user interface to information retrieval systems. We may be able to capitalize on the recent advances that have been made in ASR technology to automatically transcribe more general audio data. However, this shift toward the use of speech as a source of content from the traditional use of speech as a user interface poses new challenges to ASR technology. To understand these new challenges, one of the goals of this work is to study the general characteristics of GAD to discover it's acoustic and lexical properties, as well as the general structure of this type of data. Preliminary analysis on a collection of broadcast news data has found that GAD is characterized by disfluent speech, often containing mispronounced and incomplete words drawn from a very large vocabulary, recorded from multiple environments and speakers.

Multiple speakers may overlap in their utterances, and the speech signal may be interspersed with music and other sounds [1]. In addition, the speakers are not directly interacting with the ASR system as in the traditional use of ASR technology within the context of user interfaces. It has been shown that an ASR system does not perform uniformly well over such a variety of conditions [2].

Segmenting these data into acoustically homogeneous blocks and using appropriate models for each segment has been shown to improve overall recognition accuracies [2]. However, when examining GAD we quickly see that there are many different levels of segmentation that can be constructed. We can visualize a very coarse segmentation that indicates boundaries between speech and non-speech, another that indicates boundaries between different background acoustic environments, another that indicates boundaries between different speakers, etc. Each of these possible segmentations is useful for different reasons and should be included in any representation of GAD. However, computing such a segmentation suffers from problems with scale. For example, boundaries between very acoustically distinct regions such as silence and music must be detected, as well as boundaries between more subtly differing regions such as those between two male speakers.

Even if we consider a single level of segmentation, this scale problem exists. For example, boundaries between two speakers of different gender will presumably be easier to detect than boundaries between two speakers of the same gender. The segmentation algorithm must be tuned to locate boundaries between very different regions, and boundaries between more similar regions. Because of this problem, we feel that a single, linear segmentation could not adequately represent all of the potential segmentations that exist in the data. Therefore, in this work we propose a graph-based representation that captures many of the possible scales of segmentation that are present in GAD.

We have chosen to use a hierarchical clustering algorithm that incorporates temporal constraints to produce a multi-level acoustic description of GAD. Such a representation has been used successfully to describe speech at a phonetic level [3]. We are expanding this idea to capture the higher level acoustic regularities found in GAD. This algorithm first determines “seed regions”, which define the smallest acoustically homogeneous segments desired in the data. Then, each segment is associated with either it’s right or left neighbor using a distance measure. When two segments associate, they are merged into a new segment. The merging process continues until all segments are merged together. If we keep track of the distance at which two segments merge into one, we can display the results as a dendrogram. The bottom layers of the dendrogram structure display the fine details of GAD, while higher layers display the coarse details. Different paths through the graph yield different levels of segmentation, such as speech/non-speech, background environment, or speaker.

There are two main steps in this approach. First, we must determine how to locate the initial seed regions. One method of producing these seed regions is finding locations of maximal change in some representation of the audio signal. We are initially using the smoothed spectrum as our representation of the audio signal. We will be investigating the use of other representations that are able to capture higher level spectral change, such as that between environments or speakers, rather than locating phonetic boundaries. Second, we must determine how best to merge segments. We will be exploring a number of different feature vectors to represent each feature, as well as different distance metrics in the clustering procedure.

References:

[1] M. S. Spina, and V. W. Zue, "Automatic Transcription of General Audio Data: Preliminary Analysis," *Proc. ICSLP '96*, Philadelphia, PA, October 1996.

[2] M. S. Spina, V. W. and Zue, "Automatic Transcription of General Audio Data: Effect of Environment Segmentation on Phonetic Recognition," *Proc. Eurospeech '97*, Rhodes, Greece, September 1997.

[3] J. Glass, *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition*. Ph.D. Thesis, Massachusetts Institute of Technology, May 1988.

Mandarin Tone Sandhi and Intonation for Improved Tone Recognition

Chao Wang

Prosodic cues (namely, fundamental frequency, energy and duration) provide important information for speech. For a tonal language such as Chinese, fundamental frequency (F0) plays a critical role in characterizing tone as well, which is an essential phonemic feature. In this regard, we have focused our initial study of prosodic features on fundamental frequency (F0) contours of Mandarin Chinese. We believe that, unlike the obscure correlation of the F0 contour with stress, accent, rhythm in an intonation language such as English, the syllable level F0 contour in Mandarin clearly defines tone; and the phrase or sentence level F0 features, or intonation, can be studied within the context of improving tone recognition.

There are four lexical tones in Mandarin Chinese, each corresponding to a canonical F0 pattern: “high-level”, “high-rising”, “low-dipping” and “high-falling”. However, tones in continuous speech can vary dramatically from their canonical form, due to coarticulation from surrounding tones (tone sandhi), as well as influences from intonation. The problem of tone coarticulation, or tone sandhi, has been studied extensively for tonal languages and their dialects [1][2][3]. Shen[1] studied all possible combinations of tones of Mandarin on /pa pa pa/ tri-syllables, and found that not only the onset and offset values but also the overall heights of a tone were affected by the contexts; and the coarticulatory effects are bi-directional and symmetric. Xu [2] studied F0 contours of Mandarin bi-syllables /ma ma/ embedded in carrier sentences, and found asymmetrical bi-directional coarticulatory effects in terms of F0 onset and offset changes. Xu [3] also conducted a perceptual study of coarticulated tones and found that human performance on tone identification was highly dependent on the availability of original tonal context when the context was “conflicting” with the tone. The intonation in tone languages is less well-studied. Nevertheless, Chang [4] found that in Szechuanese (a dialect of Chinese), emphatic statements were produced with a higher pitch; various attitudes were expressed by a wider pitch range; and questions changed the utterance-final tone.

In this work, we try to use a parametric approach to characterize both the tone coarticulation and intonation effects, so that the results can be easily incorporated into automatic tone recognition. We choose Mandarin digit strings and phone numbers as an initial domain in which to begin our study, because: 1) Mandarin digits cover all four lexical tones, thus continuous digit strings provide an adequate domain to study tone contextual effects; and 2) digit strings have simple sentence and phrase level intonation structure which are relatively easy to characterize.

Tone is mainly dependent on the F0 contour pattern, i.e., its average, slope, and curvatures. There are various ways to quantify these features in a segment-based system, by either fitting the F0 contour with a certain type of function, or projecting it onto some basis functions. We have chosen to use the first four coefficients of the discrete Legendre transformation to characterize the tone contour[5]. Physically, the first coefficient corresponds to the average pitch; the second coefficient correlates with the slope; the third coefficient correlates with the second-order curvature, etc. Segment duration and average probability of voicing are also included in our tone feature vector. The duration feature does not contribute significantly to tone discrimination, but it is essential to limit insertion and deletion errors in recognition. A principle component analysis is applied to the tone feature vector, and mixtures of diagonal Gaussians are used to model the distributions. The baseline tone classification achieved an error rate of 18.8% in performance.

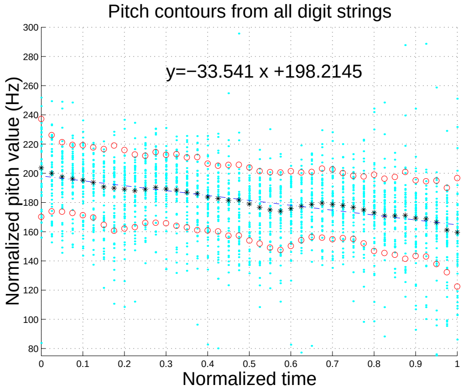


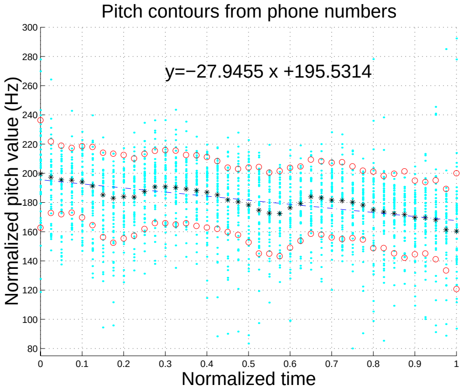
Figure 17. Pitch contours of all digit strings.

We use a "corrective" approach to account for different tonal contexts, i.e., we compensate the coarticulatory effects by "undoing" those effects, so that the resulting models will be more homogeneous. Specifically, we first train context-dependent models for each tone, then compare distributions of these context dependent models with our baseline context-independent models to obtain a set of parameters reflecting the differences. Since it is observed that the distributions of Legendre coefficients are Gaussian-like, we simply use the difference in the mean for each coefficient to characterize the contextual effects. We then alter the pitch contour of each tone according to its contexts to compensate for those differences. We then train new context-independent models from those corrected pitch contours. We performed correction for the first two Legendre coefficients and found that the variances of the new models are significantly reduced for those two dimensions, and the classification error is reduced by 11.7% from baseline.

It is generally agreed that tonal languages may make use of a limited amount of superimposed intonation. We have tried linear or quadratic regression analysis for the pitch contour of an utterance or its component phrases to approximate the intonation contour. However, the results were not very reliable, due to interference from tones as well as errors from pitch extraction. Given that our data are digit strings, we assume that the underlying intonation components are similar. So we use an "averaging" approach to smooth out the "random" variations due to tones and obtain the residue average as the underlying intonation contour.

We tested our hypothesis by plotting the pitch contours of all our data on a normalized scale, as shown in Figure 17. It is very obvious from the plot that there is a downdrift of the averaged pitch contour. A plot of pitch contours for only phone numbers shown in Figure 18 also reveals a more detailed phrase structure corresponding to "xx-xxx-xxxx". We then subtracted the sentence level intonation component from the original pitch contour and re-trained tone models, which reduced the tone classification error rate by 12.2% from the baseline. When both normalization schemes were combined, the error rate was reduced by 26.1% from baseline. More refined phrase level intonation models did not yield significant improvement over the simple sentence level model. We think this is because the phrase boundary is treated as a context in our context normalization; thus the fine structure of the intonation contour

Figure 18. Pitch contours of phone numbers.



near the phrase boundary is already accounted for in the context normalization.

We also tried our approach on spontaneous data from the Yinhe domain [6], and obtained similar improvement for tone classification. We are currently refining our method to remove the dependency on the prior knowledge of tone contexts, as well as to achieve speech recognition improvement.

References

[1] X. S. Shen, “Tonal Coarticulation in Mandarin,” *Journal of Phonetics*, Vol. 18, 281-295, 1990.

[2] Y. Xu, “Contextual Tonal Variations in Mandarin,” *Journal of Phonetics*, Vol. 25, 62-83, 1997.

[3] Y. Xu, “Production and Perception of Coarticulated Tones,” *Journal of Acoustic Society of America*, Vol. 95, No. 4, 2240-2253, 1994.

[4] N-C. T. Chang, “Tones and Intonation in the Chengtu Dialect,” *Phonetics*, Vol. 2, 59-84, 1958.

[5] C. Wang, and S. Seneff, “A Study of Tones and Tempo in Continuous Mandarin Digit Strings and their Application in Telephone Quality Speech Recognition,” *Proc. ICSLP ‘98*, Sydney, Australia, September, 1998.

[6] C. Wang. *Porting the GALAXY System to Mandarin Chinese*, M.S. thesis, Massachusetts Institute of Technology, May 1996.

Natural-Sounding Concatenative Speech Synthesis of Multi-Sonorant Sequences

Jon Yi

Recently, it has become increasingly apparent that natural-sounding speech output is highly desirable for conversational systems. Concatenative methods and unit selection mechanisms have lately received considerable attention in the speech synthesis literature [1, 2]. Our initial work in concatenative speech synthesis dealt with the concatenation of word- and phrase-level units. These units were carefully prepared by recording them in the precise prosodic environment in which they were to be used. A synthesizer, ENVOICE, was created to operate within the GALAXY architecture by converting meaning representations into speech waveforms. This overall system was capable of producing very natural-sounding synthesized speech in constrained application domains. ENVOICE was used for synthesizing responses in an air travel domain called PEGASUS. Since the vocabulary was relatively small, it was practical to record every word in every prosodic environment realizable. However, this technique may be impractical for application domains involving dynamic vocabularies or variable syntax (i.e., names of newly introduced cars in an automobile advertisement domain, or complex sentence structure of weather reports in a weather domain.) While brute-force methods would dictate the recording of every word, we decided to seek out methods that concatenate sub-word units for the synthesis of arbitrary words [3]. Our subsequent work with smaller-sized units began with perceptual studies to determine where speech could be spliced with minimal audible distortion and what contextual constraints were necessary to maintain in order to achieve naturalness during concatenation. We found splicing to be less perceptually noticeable at places of source change (e.g., unvoiced-voiced transitions.) In the final stage of our work with sub-word units, we designed a search algorithm to select a sequence of sub-word units from a speech corpus given an input pronunciation. This process of unit selection was guided by unit and transition costs that encoded contextual constraints and where speech could be spliced, respectively. These costs were organized by manner of production and place of articulation equivalence classes indexed in a triphone fashion. Because unvoiced-voiced transitions mainly coincide with phonetic boundaries, it was sufficient for our labelling of both the synthesis corpora and the unit selection search space to be purely symbolic. Provided with only phonetic alignments of the synthesis corpus, this symbolic approach was capable of producing natural-sounding synthesis for words containing single sonorant sequences, for example. Later, we found the addition of lexical stress information into the unit labels to help improve the prosody of poly-syllabic words.

Although synthesized words containing single sonorant sequences sounded natural, we noticed the tendency for poly-syllabic synthesized words containing longer sonorant sequences to sound unnatural. In cases where it was necessary to synthesize multi-sonorant sequences from smaller units, mismatch of formant structure and discontinuity in fundamental frequency at concatenation points were perceptible. From these observations it was clear that greater care was needed at these boundaries and that a purely symbolic search was not capable of capturing numerical constraints such as formant and fundamental frequency continuity between voiced regions.

More recently we have begun to

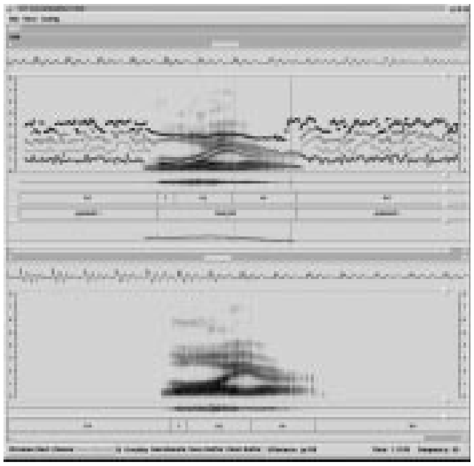


Figure 19. The Concatenation View (cv) tool is an interactive concatenation tool that allows for the cumulative construction of synthesized speech from user-selected segments of utterances in a given corpus. Note that formant tracks are overlaid onto the spectrogram and a separate plot of the fundamental frequency is provided with a following frequency cursor.

concentrate on the synthesis of contiguous multi-sonorant sequences from their constituent vowel and semivowel sounds. Since constrictions in the vocal tract typically result in energy minima, we hypothesized that splices may not be perceptible at these points and that speech sounds could be naturally concatenated at these boundaries. In Figure 19 we can see the Concatenation View (cv) tool as used in an example involving extreme articulation in diphthongs. We synthesized "lawyer" from an original "lawyer" by substituting the first syllable from "loyal," a word which shares the same syllable and overall stress pattern. The splice boundary was placed pitch-synchronously where the second spectral prominence (shown by the pink trajectory in the figure) reaches a peak in its evolution. Note that the /er/ segment selected from the original "lawyer" starts after the trajectory of the second spectral prominence levels off (marked by the light gray cursor.) When formant continuity is maintained at splice boundaries between voiced regions, we have found concatenation to introduce minimal perceptual distortion. Besides overlaying formant tracks onto a spectrogram, the tool also provides a separate plot of the fundamental frequency with a following frequency cursor. For example, F0 is 55 Hz at time 1.1109s as shown by the dark gray cursor.

Future work in the area of concatenative synthesis will involve the extension of studies to more complex polysyllabic words, phrases, and sentences in narrow domains. In the example above, concatenative synthesis was carried out manually with the aid of visual tools. In order to synthesize naturally-sounding speech in an automatic fashion, it would be necessary to use estimates of formant and fundamental frequency both in the segmentation process of corpus preparation and in the unit selection search metric. Furthermore, we have observed concatenated sonorant sequences to sound most natural when splice boundaries are placed pitch-synchronously. It may also be necessary to perform automatic detection of glottal closure instants during the segmentation process.

References

[1] A.J. Hunt and A. W. Black. "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database," *Proc. ICASSP '96*, pp. 373-376, Atlanta, GA, May 1996.

[2] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal. "The AT&T Next-Gen TTS System". Joint Meeting of ASA, EAA, and DAGA, Berlin, Germany, March 1999.

[3] J. R. W. Yi and J. R. Glass. "Natural-Sounding Speech Synthesis Using Variable-Length Units," *Proc. ICSLP '98*, pp. 1167-1170, Sydney, Australia, December 1998.

THESES

PUBLICATIONS

PRESENTATIONS

SEMINARS

Doctoral Theses

Completed

A. Halberstadt, *Heterogeneous Measurements and Multiple Classifiers for Speech Recognition*, Supervisor: J. Glass. November1998.

In Progress

K. Ng, *Subword-based Approaches for Spoken Document Retrieval.* Expected January 2000.

M. Spina, *Toward the Automatic Transcription of General Audio Data.* Expected June 2000.

G. Chung. *Towards Multi-Domain Speech Understanding with Flexible and Dynamic Vocabulary.* Expected December 2000.

Masters Theses

M.Eng Theses Completed

H.-J. Kim. *SeMole: A Robust Framework for Gathering Information from the World Wide Web*. Supervisor: L. Hetherington. September 1998.

In Progress

K. Livescu. *Analysis and Modeling of Non-Native Speech for Automatic Speech Recognition*. Supervisor: J. Glass. Expected September 1999.

S. Kamppari. *Word and Phone Level Acoustic Confidence Scoring for Speech Understanding Systems*. Supervisor: T.J. Hazen. Expected September 1999.

Publications

G. Chung and S. Seneff, “Improvements in Speech Understanding Accuracy Through the Integration of Hierarchical Linguistic, Prosodic and Phonological Constraints in the JUPITER Domain,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

J. Glass and T. Hazen, “Telephone-Based Conversational Speech recognition in the JUPITER Domain,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

A. Halberstadt and J. Glass, “Heterogeneous Measurements and Multiple Classifiers for Speech Recognition,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

H.-J. Kim and L. Hetherington, “Semole: A Robust Framework for Gathering Information from the World Wide Web,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

R. Lau, “A Unified Framework for Sublexical and Linguistic Modelling Supporting Flexible Vocabulary Speech Understanding,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

S. Lee and J. Glass, “Real-Time Probabilistic Segmentation for Segment-Based Speech Recognition,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

K. Ng, “Towards Robust Methods for Spoken Document Retrieval,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

C. Pao, P. Schmid and J. Glass, “Confidence Scoring for Speech Understanding Systems,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

S. Seneff, “The Use of Linguistic Hierarchies in Speech Understanding,” *Invited Keynote paper, Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

S. Seneff, et al. “GALAXY-II: A reference Architecture for Conversational System Development,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

C. Wang and S. Seneff, “A Study of Tones and Tempo in Continuous Mandarin Digit Strings and their Application in Telephone Quality Speech Recognition,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

J. Yi and J. Glass, “Natural Sounding Speech Synthesis Using Variable Length Units,” *Proc. ICSLP ‘98*, Sydney, Australia, December 1998.

J. Glass, T.J. Hazen and L. Hetherington, “Real-time Telephone-Based Speech Recognition in the JUPITER Domain,” *Proc. ICASSP ‘99*, Phoenix, AZ, March 1999.

G. Chung and S. Seneff, “A Hierarchical Duration Model for Speech Recognition Based On the ANGIE Framework,” *Speech Communication* 27 (1999) 113-134.

Presentations

V. Zue, "The Next Generation of Speech-based Interfaces: from Recognition to Conversation," ASAT '98, San Francisco, CA, July 28, 1998.

J. Glass, "Conversational Speech Understanding Interfaces for Human Computer Interaction: Applications of Dynamic Programming and A*-Optimal Algorithms," invited presentation at the joint meeting of IEEE Control Systems, Signal Processing and Computer Science Societies, September 16, 1998.

V. Zue, "JUPITER: A Telephone-Only Interface to On-Line Weather Information ITRI, Taipei, Taiwan, September 17, 1998.

J. Glass, "Natural Sounding Speech Synthesis for Conversational Interfaces" invited lecture at Fidelity Investments, September 29, 1998.

S. Seneff, "GALAXY Update," Lucent, New Jersey, October 23, 1998.

V. Zue, "Research and Development of Conversational Interfaces," Intel Beijing Research Center opening, Beijing, China, November 4, 1998.

V. Zue, S. Seneff and R. Lau," GALAXY-II: A reference Architecture for Conversational System Development," DARPA Communicator Meeting, Pittsburgh, PA, November 12, 1998.

V. Zue, "The Next Generation of Speech Based Interfaces," Philips, November 27, 1998.

S. Seneff, "The Use of Linguistic Hierarchies in Speech Understanding," Keynote address, ICSLP'98, Sydney, Australia, December 4, 1998.

S. Seneff, "GALAXY-II: A Reference Architecture for Conversational System Development," DARPA Communicator Kick-Off meeting, San Diego, January 13, 1999.

V. Zue, "They Talk Back Too! The Coming Revoluation of Speech Technology," World Economic Forum Annual Meeting, Davos, Switzerland, February 2, 1999.

V. Zue, "Multilingual Conversational Interfaces," NTT, March 23, 1999.

V. Zue, "An Overview of Aspects of Spoken Language Technology," Intel Speech Symposium, Hillsboro, OR, April 1, 1999.

V. Zue, "Speech Is It," LCS-35 Anniversary Celebration, Cambridge, MA , April 13, 1999.

S. Seneff, "The Use of Linguistic Hierarchies in Speech Understanding," invited talk at Universidad de las Americas, Puebla, Mexico, April 7, 1999.

V. Zue, "The Next Generation of Speech Based Interfaces," invited talk at Universidad de las Americas, Puebla, Mexico, April 7, 1999.

J. Glass, "Natural Sounding Synthesis using Variable-Length Units," Invited seminar at Boston University, Boston, MA, April 26, 1999.

Presentations

V. Zue, DARPA Communicator: Architecture, Applications, & Issues, XIWT, May 4, 1999.

V. Zue, “Speech Is It,” EPOCH-IT Symposium, Taipei, Taiwan, May 26, 1999.

J. Glass, “Acoustic Properties of Speech Sounds,” invited lecture at Johns Hopkins University, Baltimore, MD, June 29, 1999.

SLS Seminar Series

October 6, 1998
Bridge the Gap Between Humans and Machines for Better Communication
X.D. Huang
Microsoft Research

October 14, 1998
Automatic Methods for Building Natural Language Understanding Systems
Salim Roukos
IBM T. J. Watson Research Center

October 20, 1998
Dialogue Systems for Interactive Problem Solving
James Allen
University of Rochester

Tuesday, November 3, 1998
Concept to Speech Generation for Multimedia Briefings
Kathleen R. McKeown
Columbia University

February 23, 1999
Improving Accent Identification through Knowledge Of English Syllable Structure
Kay Berkling
MIT Lincoln Laboratory

April 30, 1999
The Erlangen Spoken Dialogue System EVAR: A State-of-the-Art Information Retrieval System
Elmar Noeth
University of Erlangen-Nuremberg

Photography by Tony Rinaldo

1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999 1989-1999 1989-1999

1989-1999 1989-1999 1989-1999

MIT
Massachusetts
Institute
of Technology