Leveraging Multimodal Redundancy for Dynamic Learning, with SHACER — a Speech and HAndwriting reCognizER

Edward C. Kaiser
B.A., American History, Reed College, Portland, Oregon (1986)
A.S., Software Engineering Technology, Portland Community College, Portland, Oregon (1996)
M.S., Computer Science and Engineering, OGI School of Science & Engineering at Oregon Health & Science University (2005)

A dissertation submitted to the faculty of the
OGI School of Science & Engineering at
Oregon Health & Science University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Computer Science and Engineering

April 2007

The dissertation "Leveraging Multimodal Redundancy for Dynamic Learning, with SHACER — a Speech and HAndwriting reCognizER" by Edward C. Kaiser has been examined and approved by the following Examination Committee:

Philip R. Cohen
Professor
Oregon Health & Science University
Thesis Research Adviser

Randall Davis
Professor
Massachusetts Institute of Technology

John-Paul Hosom
Assistant Professor
Oregon Health & Science University

Peter Heeman
Assistant Professor
Oregon Health & Science University

Dedication

To my wife and daughter.

Acknowledgements

This thesis dissertation in large part owes it existence to the patience, vision and generosity of my advisor, Dr. Phil Cohen. Without his aid it could not have been finished.

I would also like to thank Professor Randall Davis for his interest in this work and his time in reviewing and considering it as the external examiner. I am extremely grateful to Peter Heeman and Paul Hosom for their efforts in participating on my thesis committee.

There are many researchers with whom I have worked closely at various stages during my research and to whom I am indebted for sharing their support and their thoughts. I would like to thank Ron Cole for his early support of my desire to study computer science, and Wayne Ward for many valuable discussions about parsing and grammar compilation early in my time as a doctoral student. During my work on the CALO project I have been blessed with many fruitful collaborations and I would like acknowledge my gratitude to Alex Rudnicky and Alan Black for their willingness to help in the development of SHACER. Also I'm grateful to David Demirdjian for his unfailing friendship and support in many related research endeavors.

I have enjoyed being part of both the Center for Human Computer Communication (CHCC) and now of Adapx. during my years as a PhD student. My thanks to Paulo Barthelmess, Xiao Huang, Matt Wesson, Xioaguang Li, Sanjeev Kumar, David McGee, Michael Johnston, Andrea Coradini and others for their efforts over the years without which my work and SHACER in particular would not have been possible. To my fellow PhD students, Rebecca and Raj, I wish all the best for their forthcoming theses.

Finally, I would like to thank my wife and daughter for their trust and love over these last years.

I am grateful to the Defense Advanced Research Projects Agency (DARPA) for financially supporting my thesis research through the Contract No. NBCHD030010 for the CALO program. I note that any opinions, findings and conclusions or recommendations expressed in this material are my own and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC). I would also like to thank SRI for letting me work on parts of my thesis as part of the CALO project.

Contents

List of Tables

List of Figures

Abstract

Leveraging Multimodal Redundancy for Dynamic Learning, with SHACER — a Speech and HAndwriting reCognizER

Edward C. Kaiser

Supervising Professor: Philip R. Cohen

New language constantly emerges from complex, collaborative human-human interactions like meetings — such as when a presenter handwrites a new term on a whiteboard while saying it redundantly. Fixed vocabulary recognizers fail on such new terms, which often are critical to dialogue understanding. This dissertation presents SHACER, our Speech and HAndwriting reCoginzER (pronounced "shaker"). SHACER learns out-of-vocabulary terms dynamically by integrating information from instances of redundant handwriting and speaking. SHACER can automatically populate an MS Project ^{TM} Gantt Chart by observing a whiteboard scheduling meeting.

To document the occurrence and importance of such multimodal redundancy, we examine (1) whiteboard presentations, (2) a spontaneous brainstorming meeting, and (3) informal annotation discussions about travel photographs. Averaged across these three contexts 96.5% of handwritten words were also spoken redundantly. We also find that redundantly presented terms are (a) highly topic specific and thus likely to be out-of-vocabulary, (b) more memorable, and (c) significantly better query terms for later search and retrieval.

To combine information SHACER normalizes handwriting and speech recognizer outputs by applying letter-to-sound and sound-to-letter transformations. SHACER then uses an articulatory-feature based distance metric to align handwriting to redundant speech. Phone sequence information from that aligned segment then constrains a second pass phone recognition over cached speech features. The resulting refined pronunciation serves as a measure against which the integration of all orthographic and pronunciation hypotheses is scored. High-scoring integrations are enrolled in the system's dictionaries and reinforcement tables. When a presenter subsequently says a newly enrolled term it is more easily recognized. If an abbreviation is handwritten at the same time, then the already recognized spelling is compared to the handwriting hypotheses. If there is a first-letter or prefix match, then that full spelling is dynamically acquired by the handwritten abbreviation as its expanded meaning. On a held-out test set SHACER significantly reduced the absolute number of recognition errors for abbreviated Gantt chart labels by 37%.

For cognitive systems to be accepted as cooperative assistants they need to learn as easily humans. Dynamically learning new vocabulary, as SHACER does by leveraging multimodal redundancy, is a significant step in that direction.

Chapter 1

Introduction

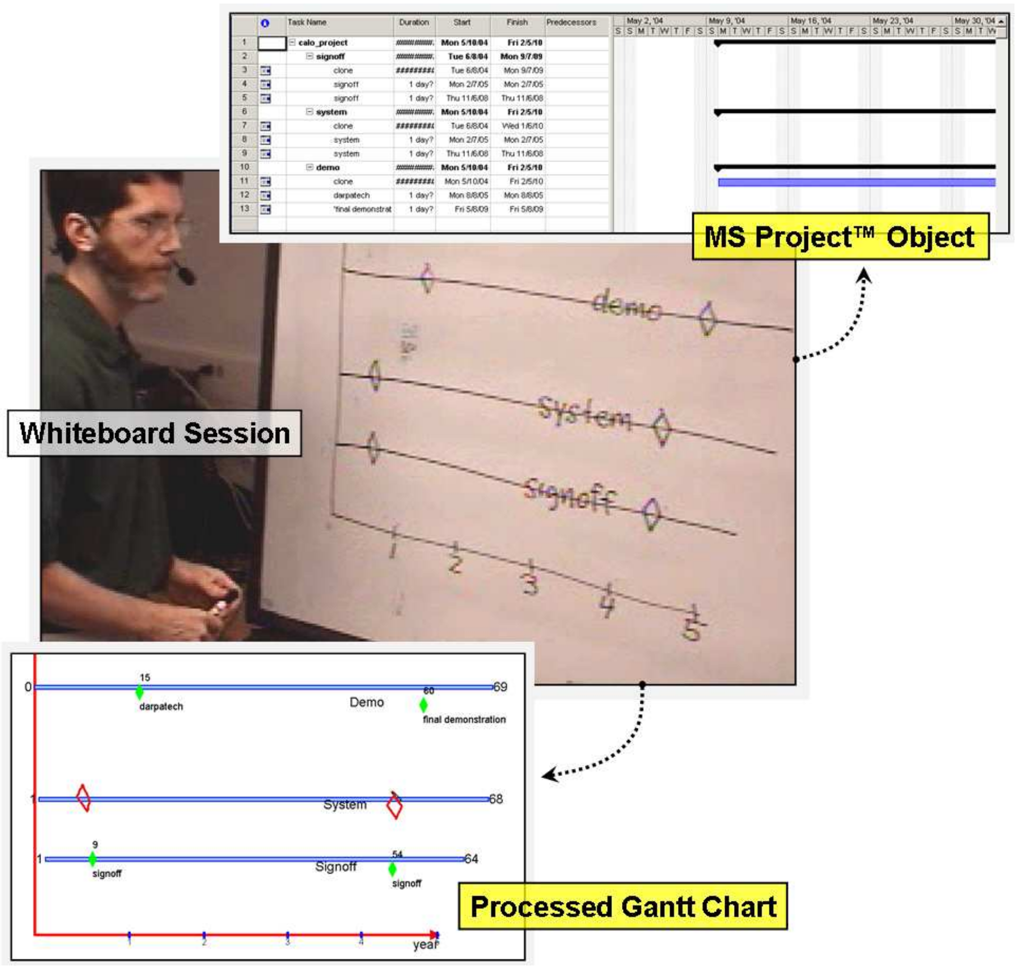
1.1 MOTIVATION

As computers become perceptually more capable, new types of computational systems are becoming feasible. It is now possible to conceive of computational systems that understand our regular communicative lives without requiring that we sit in front of a computer to enter input [37]. For example, during meetings people present information to each other across multiple modes. Graphically, they sketch diagrams, like a schedule chart or timeline. Textually, they handwrite lists of important points, or they label parts of their diagrams. While they are sketching or handwriting they are also speaking to each other. Speakers may handwrite on public surfaces (like whiteboards, flipcharts or even table napkins), while listeners jot down personal notes on paper. In the background of such meetings computers can unobtrusively act as ambient perceptual agents. They can observe the speech, handwriting, and sketching communication occurring between people. Speech can be recorded through microphones and then recognized by speech recognition systems [60, 118]. Similarly, handwriting and sketching can be perceived through various ink-sensitive surfaces and then recognized by sketch or handwriting recognizers [4, 6].

As ambient perceptual computers become more effective at understanding peoples' various communicative modes, they can better serve people's needs by performing useful background services. Instead of sitting in front of a computer to access these services, people could simply interact freely with each other [37, 165]. Advances in background perceptual understanding could dramatically enhance business productivity, for example, by better facilitating meeting collaboration and summarization [4]. Such advances could also make it easier to search for and recover aspects of archived human-human interactions recorded in computationally perceptual settings [106].

This thesis argues that combining information from speech and handwriting can help computers do a better job at background understanding of complex human-human interactions like meetings or lectures. Evidence offered in this thesis will show that, during meetings or presentations, handwritten words are typically also spoken redundantly. The information available in the redundant speech and handwriting can be combined to produce recognition that is significantly more accurate than the recognition achievable in either mode alone. This improved accuracy means that more of the handwritten or spoken words can be recognized correctly. Moreover, during meetings important words are often out-of-vocabulary [179]. This is a significant problem for computer understanding, because out-of-vocabulary words during meetings tend to be important words [179, 181], with as much as 70% of out-of-vocabulary words being named-entities like proper names [3]. When a name is out-of-vocabulary it cannot be recognized. Instead, the recognizer substitutes other words in its place, which corrupts the recognizer's word sequence modeling and causes a cascade of local recognition errors in the vicinity of the substitution.

Current computer recognition systems, unlike people, cannot enroll new vocabulary just by hearing it spoken or seeing it written in context. Commercial speech recognizers revert to explicitly asking users to type in the spelling of a new word while also speaking it, so that the system can specifically add the new spelling and pronunciation to its vocabulary. The system presented in this thesis is a background understanding system. It does not explicitly ask for user supervision and guidance to enroll new vocabulary. Instead, since handwriting is often spoken redundantly during meetings or lectures, it leverages that naturally occurring multimodal redundancy as a basis for enrolling new words. Multimodal redundancy means that the same information is presented in more than one mode, like handwritten words that are also spoken. Multimodal redundancy provides implicit supervision for enrolling new words — the spelling of a newly spoken word is determined from the corresponding redundant handwriting, while the pronunciation of a newly handwritten word is extracted from the corresponding redundant speech. Thus, this work argues that multimodal redundancy is the fulcrum on which understanding new words in context can be leveraged by computational systems.



		Task Name	Duration	Start	Finish	Predecessors
1		calo_project	##########.	Mon 5/10/04	Fri 2/5/10	
2		signoff	##########.	Tue 6/8/04	Mon 9/7/09	
3		clone	#########	Tue 6/8/04	Mon 9/7/09	
4		signoff	1 day?	Mon 2/7/05	Mon 2/7/05	
5		signoff	1 day?	Thu 11/6/08	Thu 11/6/08	
6		system	##########.	Mon 5/10/04	Fri 2/5/10	
7		clone	#########	Tue 6/8/04	Wed 1/6/10	
8		system	1 day?	Mon 2/7/05	Mon 2/7/05	
9		system	1 day?	Thu 11/6/08	Thu 11/6/08	
10		demo	##########.	Mon 5/10/04	Fri 2/5/10	
11		clone	#########	Mon 5/10/04	Fri 2/5/10	
12		darpatech	1 day?	Mon 8/8/05	Mon 8/8/05	
13		'final demonstrat	1 day?	Fri 5/8/09	Fri 5/8/09	

Figure 1.1: A whiteboard scheduling session (center), processed unobtrusively in the session's background by Charter, our prototype perceptual multimodal scheduling application, showing the dynamically evolving Gantt chart display (lower left) along with its immediately derived MS Project ^{TM} chart object (upper right).

This dissertation introduces a prototype implementation of an ambient, perceptual system, that can track the communicative interactions between people across various modes like speech, sketching and handwriting during a business meeting. Our system is named SHACER (pronounced *shaker*), which is an acronym for Speech and HAndwriting reCognizER. The presenter in Figure 1.1 is drawing a schedule chart with task-lines spanning the duration of the project and specific goals on each task-line sketched in as diamond-shaped milestones. During processing of the meeting the system continually interprets and updates its understanding of the interaction. The system's understanding is reflected as both a beautified version of the labeled Gantt chart (Figure 1.1, lower left) and as an automatically populated Microsoft Project ^{TM} object (shown in the upper right of the illustration). The information in the Microsoft Project ^{TM} object is the same as that reflected in the beautified Gantt chart, and includes a lexicon of expanded abbreviation and acronym meanings, learned by the system dynamically as it observed and processed the meeting.

SHACER can automatically detect occurrences of multimodal redundancy across handwriting and speech. It can then combine information across those redundant modes to dynamically discover the spelling and pronunciation of new words or phrases. SHACER's enrolling those dynamically discovered new words into the systems' dictionaries and language models constitutes learning. For SHACER learning specifically means acquiring new vocabulary, which is a critical part of acquiring or learning language. This definition fits well with the definition of learning given by Herbert Simon, who defined learning as, "changes in a system that result in improved performance over time on tasks similar to those done previously" ([50], pg. 600). This definition suggests that continuous, cumulative improvement is the acid test of learning. Performance results for the system described in this thesis show significant, cumulative improvement across a series of test meetings. SHACER learns new vocabulary in early meetings and then uses that enrolled vocabulary to improve recognition in later meetings.

In order to better understand human-human interactions, computational systems need to be generally more able to combine recognitions across various modes of perceptual input to make dynamic, unsupervised inferences about how multimodal information should be meaningfully combined. For example, they should be able to infer that an email dragged to a folder belongs to that folder and therefore has some semantic relationship to other emails in that folder. They also be able to infer that the handwritten letters of new word should be associated with the redundantly spoken pronunciation for that same word. SHACER makes these sort of dynamic, unsupervised inferences in processing multimodal redundancy. The Defense Advanced Research Projects Agency's Cognitive Assistant that

Learns and Organizes [29]) (CALO) is predicated on the belief that such learning capabilities may ultimately support artificially intelligent systems that can respond robustly to surprising or unforseen inputs, just as people do. CALO's aim is to transform computational systems from being simply reactive to being truly cognitive [24]. In order for a computational system to be truly cognitive it must at least be able to learn on its own. This thesis argues that SHACER, by leveraging naturally occurring multimodal redundancy to enroll new words, demonstrates the cognitive ability to learn. The learning ability demonstrated by SHACER is an important step towards achieving CALO's aim of truly cognitive systems.

1.2 THE PROBLEM: NEW LANGUAGE AND THE NEED FOR DYNAMIC LEARNING

As machines move closer to being observant and intelligent assistants for humans [119, 25, 70, 26, 164, 126, 9, 23] it is not enough that they rely on off-line models for the support of recognition. They need to automatically adapt and acquire new models and new knowledge as they are running [9], particularly by a single, natural demonstration. Machines or systems that assist humans in real-time tasks need to be able to learn from being shown — through sketch [38, 154], handwriting [100], teleassistance [143], speech [162], or multimodally (as in the work described in this dissertation) through handwriting and speech.

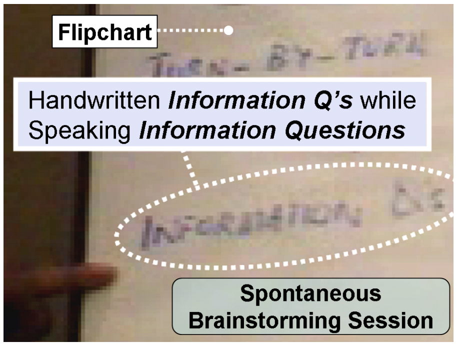

Flipchart
TURN-BY-TURN
Handwritten *Information Q's* while Speaking *Information Questions*
INFORMATION Q'S
Spontaneous Brainstorming Session

Figure 1.2: Introduction of a new abbreviation during a ninety minute spontaneous brainstorming session, using a whiteboard and flip chart.

New language and new language forms constantly emerge from complex, collaborative human-human interactions like meetings and presentations. For instance, presenters often handwrite new terms on a whiteboard or flipchart while saying them. In Figure 1.2 the presenter writes the phrase, *Information Q's*, while saying, "[. . .] information questions". This action grounds the abbreviation, *Q*, to its expanded word form, *Question*. That grounding has the effect that for the remainder of the presentation, when the presenter writes the phrase *information Q*, it will readily be understood by other participants to mean *Information Question*.

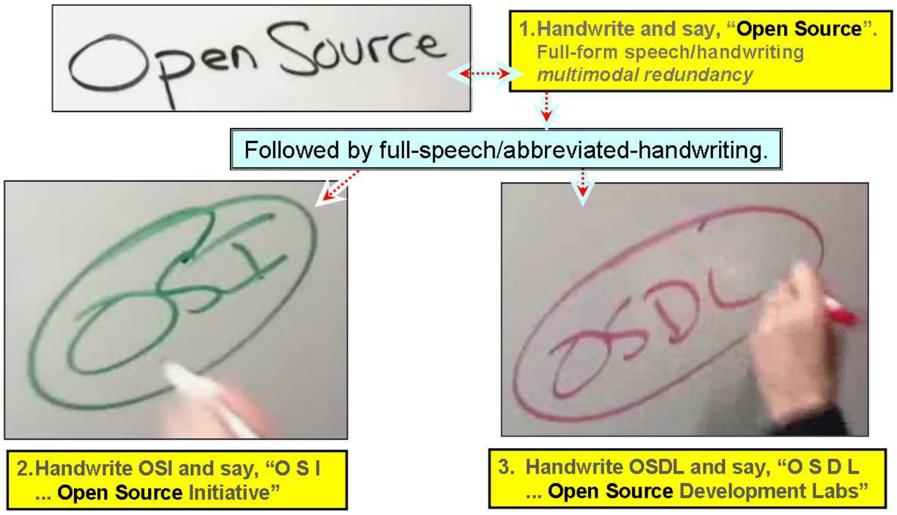
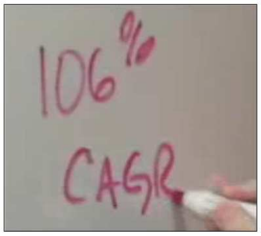

Open Source
1.Handwrite and say, "Open Source".
Full-form speech/handwriting
multimodal redundancy
Followed by full-speech/abbreviated-handwriting.
OSI
OSDL
2.Handwrite OSI and say, "O S I
... Open Source Initiative"
3. Handwrite OSDL and say, "O S D L
... Open Source Development Labs"

Figure 1.3: An example of redundantly presented, dialogue-critical terms from an online presentation [182]: a full, multimodally redundant term introduction (e.g. *Open Source*, top-left), followed by related handwritten abbreviations that are spoken in full.

A second example of new terms being dynamically introduced is shown in Figure 1.3. The presenter begins by introducing the phrase, *Open Source*, which is both handwritten and spoken. Because it is a topic-specific phrase it may not exist in the system's vocabularies, and without the attentional focus occasioned by its multimodal introduction (see Section 2.1.4) it might also not exist in the forefront of the presentation observers' minds. The presenter then subsequently introduces several new acronyms, *OSI* (for *Open Source Initiative*) and *OSDL* (for *Open Source Development Labs*). These acronyms both begin with first-letter abbreviations for the phrase, *Open Source*, and thus they are more readily understood in the context of their relation to that already grounded term. People can dynamically recognize and process this kind of grounding; however, a handwriting recognizer, with a fixed, off-line vocabulary and static word sequence model, would fail to recognize *OSI*, because it is likely to be out-of-vocabulary and not part of the recognizer's word sequence model. Fixed vocabulary recognizers, with static, off-line language models, fail on such newly created terms. These terms, as this example shows, are often critical to dialogue understanding.



Typically:
CAGR = Compound Annual Growth Rate

Idiosyncratically defined here:
CAGR = Category Growth Rate

Figure 1.4: In this example the presenter handwrites *CAGR* while saying *Category Growth Rate*.

The importance of this contextually grounded relation to a redundantly introduced term can be clearly seen in the event depicted in Figure 1.4. The presenter introduces the abbreviation *CAGR* while saying *Category Growth Rate*. The five top page hits of a Google search on the abbreviation *CAGR* give the expanded meaning of *CAGR* exclusively as *Compound Annual Growth Rate*. Thus, relying on a static dictionary of common abbreviations, as might be compiled from top Google page hits for common abbreviations, would lead to an incorrect interpretation of *CAGR* = *Compound Annual Growth Rate*. To find the correct interpretation for *CAGR* the dynamic multimodal context is needed. The presenter's redundant speech holds the key to the correct interpretation of *Category Growth Rate*. The same could be true for the abbreviation *OS* in the Figure 1.3 example, which by static dictionary lookup could easily mean any of *Ordnance Survey*, *Operating System*, or *Office of the Secretary*.

For the example in Figure 1.3 not only is *Open Source* first introduced redundantly across handwriting and speech, but it is then also used repeatedly in varying forms during the subsequent presentation. Multimodal context helps to define its correct expansion, and the frequent repetition of that context is *de facto* evidence of the term's dialogue criticalness. We will show later in this dissertation that the occurrence of multimodal redundancy, e.g. handwriting a phrase like *Open Source* and also saying it, can be successfully tracked by a computational system and leveraged as a means of enrolling new vocabulary in the system's recognizers (Chapters 5, 6 and 7). Subsequent speaking of *Open Source* while handwriting its first-letter abbreviation can then be automatically recognized, and the grounding between the initial letters, *OS* (which prefix both *OSI* and *OSDL*) and its associated phrase, *Open Source*, can be made explicit as part of an observant multimodal system's background understanding of such a presentation.

1.3 MULTIMODAL REDUNDANCY AS A BASIS FOR DYNAMIC LEARNING

Multimodal redundancy occurs when the information in one input mode is semantically the same as information in another input mode, as for example, when a presenter handwrites a phrase like, "Propose your solution," while also saying it, as shown in Figure 1.5.

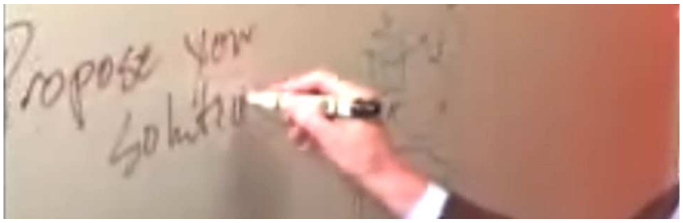


Figure 1.5: Multimodal Redundancy across handwriting and speech: a whiteboard presenter handwriting *Propose your solution* while also saying, "[. . .] Propose your solution."

Observing and recognizing redundancy in rich multimodal environments could provide the threshold ability a cognitive machine requires to allow fully bootstrapped learning. By bootstrapped learning we mean learning that requires no external supervision, that leverages the system's current capabilities in order to expand and refine its future capabilities, and that allows the system to improve on its own over time and usage.

An apt example of fully bootstrapped learning is human language acquisition. Language is composed of symbols, which in turn are grounded in perception [30, 68]. In order

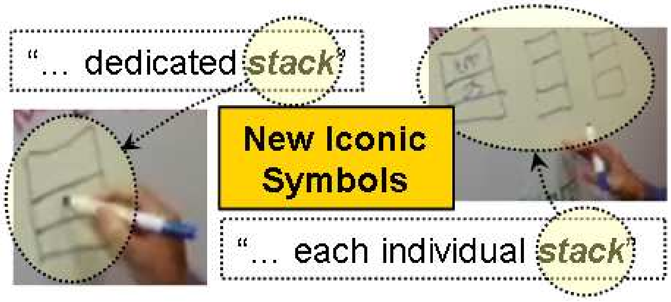


Figure 1.6: A new *stack* sketch symbol being iconically grounded via speech and sketch.

for machines to accomplish the same kind of bootstrapped learning they need to be able to ground perceptual features to recognizable symbols. We propose that multimodal redundancy can be used by machines to ground new language in this way. Some of the ways in which redundant multimodal information may provide a basis for dynamic machine learning are exemplified in the following perceptual environments:

- Redundant speech and 2D sketch could support dynamic enrollment of new sketch objects, by grounding of iconic sketches to spoken language. For example, in Figure 1.6 a presenter says, "dedicated stack," while redundantly sketching an iconic *stack* diagram. Later, after sketching several more similar *stack* icons, he references them with a deictic point gesture, and as he's gesturing again redundantly uses the word, *stack*, to describe them.

- Redundant speech and 3D gesture could support dynamic enrollment of new manipulative or iconic 3D gestures. For example, in Figure 1.7, the user makes a required but awkward manipulative hand/wrist gesture while saying, "Flip that chair." Leveraging multimodal redundancy could support the grounding of a new (perhaps simpler or more natural gesture) to the semantics of the spoken command. This sort of grounding could apply as well to new head/body posture significations of assent/dissent or attention/inattention.

- Redundant gaze, speech and face recognition could support dynamic enrollment of new faces in a face recognition module installed as part of a large meeting understanding system.

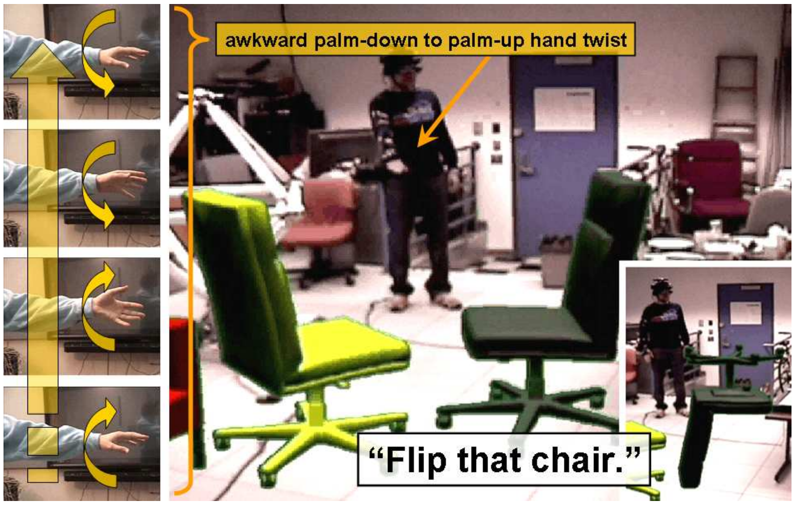


Figure 1.7: Manipulative grounding via an awkward 3D gesture and speech.

- Redundant speech, gaze and visual activity recognition could support dynamic enrollment of new activity types.

All of these contexts of multimodal redundancy lend themselves to the task of learning new vocabulary — either spoken, sketched, handwritten or gestural vocabulary. Collectively we refer to techniques for computationally facilitating this type of grounding as Multimodal Out-Of-Vocabulary Recognition (MOOVR). In implementing SHACER, we have operationalized the notions underlying MOOVR.

1.4 LEVERAGING MULTIMODAL REDUNDANCY

Our first step towards a full Multimodal Out-Of-Vocabulary Recognition system that could ultimately learn as easily as humans has been to create a prototype cognitive system that learns implicitly from the observation of redundant handwriting and speech. Speech and handwriting recognizers are closely related, relatively effective and inexpensive, and as such lend themselves particularly well to the support of a system that learns dynamically over time.

Others have worked with combining information from vision-based object recognition and speech as a basis for learning [150, 147], but vision-based object recognition is still relatively immature and expensive. Speech and handwriting are also symbolic languages and thus allow for meaningful phrases to be built up from combinations of lower level symbols. In building up such combinations not all lower level symbols need occur in the same mode. It is possible, by leveraging multimodal redundancy (as we will show in this dissertation), to transfer knowledge about learned symbols in one mode to unknown symbols in another mode (Sections 6.4.2) — as shown in Figure 1.3, where the *OS* prefix of the unknown handwritten symbols, *OSI* and *OSDL*, is assigned the learned pronunciation and expanded spelling of a known spoken phrase, *Open Source.*

1.4.1 Problems to be addressed

This thesis is concerned with the fact that during human-human interactions, when people write words in a public space — like a whiteboard or tablet PC surface displayed on a projection screen — they typically also say what they handwrite. Such events are exemplary instances of multimodal redundancy. The problem we address first is to gain some understanding of why, when and how such multimodal redundancy occurs in human-human communication (see Chapter 2). Then given some understanding of the occurrence and significance of multimodal redundancy we address the problem of designing and implementing an observant multimodal system that can leverage such redundancy to support dynamic learning of new words and abbreviations (see Chapters 4, 5, 6 and 7).

1.4.2 Hypothesis

Our hypothesis is that during multiparty interactions, where there is a shared space for writing, the participants' public handwriting will be accompanied by redundant speech. We further claim that a properly designed perceptual system can observe such human-human interactions and based on the occurrence of multimodal redundancy across handwriting and speech dynamically learn new words and abbreviations. We believe that the ability to observe natural human-human interactions across handwriting and speech and perform useful background understanding is a significant step forward toward truly cognitive machines that can assist humans.

1.4.3 Argument

Our thesis argument has two parts. First, we argue that multimodal redundancy across handwriting and speech not only occurs non-trivially but is in fact typical of some types of human-human interactions. We report on several empirical explorations we have conducted that confirm the occurrence of multimodal redundancy across handwriting and speech, and illustrate some important properties of such redundantly presented words — namely that they tend to be dialogue-critical words as evidenced by their significantly higher *term frequency — inverse document frequency (tf-idf)* weights (Chapter 2).

Second, we frame the import of those empirical findings by describing the design and implementation of our Speech and HAndwriting reCognizER (SHACER) (Chapters 5 and 6). In Chapter 7 we discuss test results that confirm the efficacy of combining handwriting and speech to better model the spelling and pronunciation of new terms, and we test SHACER's learning capabilities directly on a held-out test set of meetings. We describe how SHACER leverages the occurrence of multimodal redundancy to dynamically learn new words and thus significantly improve recognition of Gantt chart constituent labels, which happen to be out-of-vocabulary abbreviations. Abbreviation label errors are significantly reduced by 37% absolute. Thus, our prototype system significantly improves its understanding of interactions during the test meetings, due to dynamic learning of redundantly presented new vocabulary. On out-of-vocabulary data not previously seen by the system, SHACER's ability to recognize new terms improves on its own, without supervision.

Chapter 2

Empirical Studies of Multimodal Redundancy

As described in the Introduction, SHACER is designed to learn dynamically from instances of multimodal redundancy across handwriting and speech; but, how often does such redundancy occur? The literature on multimodal command systems suggests that it hardly ever happens [133], while the literature on multimedia learning suggests that it may be typical [6, 5]. Which is true? If multimodal redundancy were only to happen very infrequently, or if it happened only in situations where there was no need for computational background understanding, then developing SHACER would not be important or relevant. Therefore, before describing SHACER in detail, this chapter first reports the results of our examinations of the frequency and importance of multimodal redundancy.

This chapter argues that redundantly presented words, i.e., words that are both handwritten and spoken:

- are typical of multi-party settings where handwriting is intended to be seen publicly as part of a multimodal interaction dialogue,
- are dialogue-critical for understanding the interactions in which they occur, and
- are also likely to be highly situation-specific words and thus out-of-vocabulary.

This means that SHACER's ability to learn redundantly presented words is doubly important, because the words it can learn are both critical for understanding and likely to be out-of-vocabulary. Without the dynamic learning of new words, which SHACER can provide, background understanding of natural interaction contexts like presentations or meetings would be significantly impaired.

2.1 A WORKING HYPOTHESIS OF MULTIMODAL REDUNDANCY

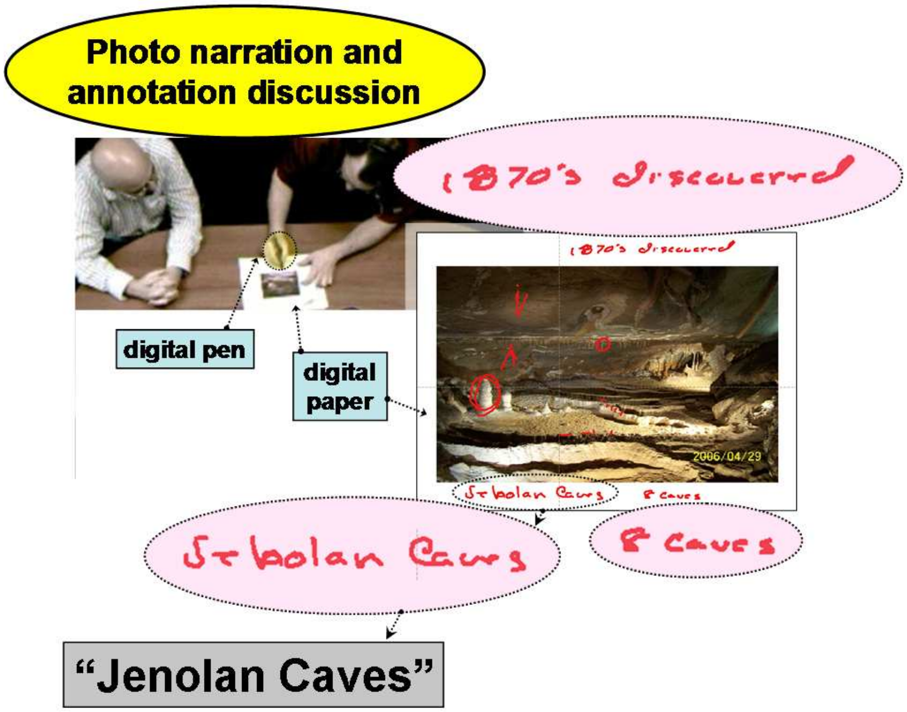


Figure 2.1: In narrating a travelogue about photos printed on digital paper, the handwriter labels the place name, *Jenolan Caves*, while also saying, "...this is the Jenolan Caves." This interaction, unlike earlier whiteboard examples, occurs in a much less formal setting where the informal public space is a shared piece of paper.

In multi-party interactions humans use multiple modes of communication in predictable ways. Grounding, for example, is the process by which we attach meaning to symbols we create [68], like handwriting a place's name below its image in a photo while talking about it as shown in Figure 2.1. Lexical entrainment [27] is the process of collaboratively arriving at dialogue-critical terms, which are shared references to the objects under discussion. For example, speaking the phrase, "Information Questions," in full while handwriting its abbreviation, *Information Q's*, on a flipchart (Figure 2.2) during a brainstorming session, and then inviting implicit acknowledgement by briefly pausing or glancing at other participants, serves to entrain the use of the abbreviation, *Information Q's*, to subsequently mean *Information Questions*.

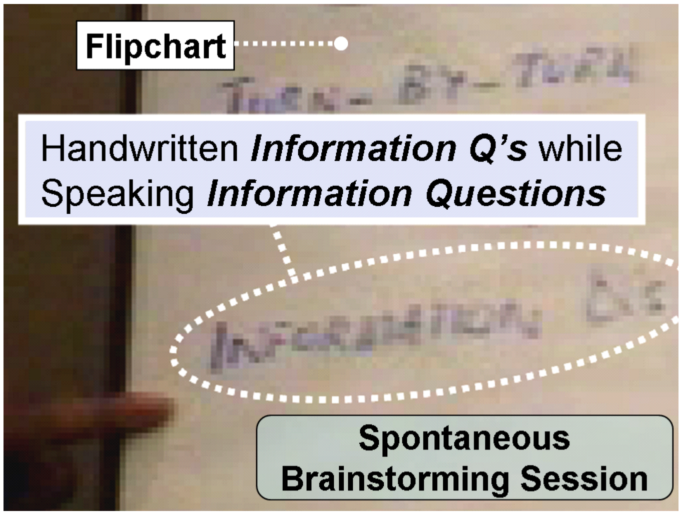


Figure 2.2: Entraining the meaning of the abbreviation, *Information Q's*, by saying "information questions" while handwriting it on a flipchart during a spontaneous brainstorming session.

It has been argued that humans expend all and only the necessary conversational energy to accomplish communication [63, 35]. Part of communication is dialogue grounding and entrainment. Herbert Clark's Principle of Least Collaborative Effort [43, 42] argues that dialogue partners will try to minimize the collaborative effort it takes to reach a level of understanding. It is clear that multimodal redundancy — e.g., both handwriting and speaking a term — requires more energy than unimodal communication alone. Therefore, there must be important communicative purposes driving its use. We believe that purpose is establishing and entraining a common ground of meaning. Our hypothesis is that people use redundancy as a conversational strategy to bolster their communicative effectiveness by drawing attention to the meanings of dialogue-critical terms. This working hypothesis is suggested by the literature on early language acquisition, which points to the importance of multimodality for focusing attention [61, 12] and thus providing a basis for understanding intentionality and establishing the meaning associations between action and intention that ground language [13, 14, 108, 168, 171, 177, 15]. We have adopted this hypothesis as a first step towards understanding why and how people use multimodal redundancy. To begin establishing empirical support for this hypothesis, we derive and consider two claims: (1) if multimodal redundancy is a general conversational strategy then it should be typical of human-human interaction settings where multiple modes can be perceived, and (2) if redundantly presented terms are dialogue-critical then they should be measurably more important than other words. In this chapter we will prove both of these initial claims.

2.1.1 Multimodal Complementarity Versus Redundancy

In multimodal command systems, like those designed for in-car navigation [66, 65], real-estate queries [129, 184, 139], emergency management [134, 107], military applications [44, 47], or pedestrian navigation [78], prevailing knowledge holds that multimodal redundancy occurs only for between 1-5% of interactions [136, 65]. Thus, as Oviatt [131] has pointed out, empirical studies of such multimodal command interfaces show that complementarity and not redundancy is the major organizational theme [133]. However, in very recent work on a human-computer command scenario for Traffic Incident Management, Ruiz *et al.* [120] found that users issued redundant commands between 30%-60% of the time. Users were given a choice of issuing commands with either speech, iconic hand gestures, or redundantly using both speech and gesture together.

For human-human interactions, Anderson *et al.* [6, 5] have shown that during distance-learning lectures 15% of ink strokes were handwriting and 100% of that handwriting was accompanied by semantically redundant speech. These distance-learning lectures were computer-mediated by the use of a tablet PC equipped with special presentation software. Thus, it appears that multimodal redundancy is less frequent during human-computer interaction scenarios than during human-human, computer-mediated scenarios like lectures.

In this chapter we will confirm Anderson *et al's* findings with results from three new empirical explorations (Section 2.2): (1) an exploration of online whiteboard presentations, (2) an exploration of a spontaneous brainstorming session, and (3) an exploration of discussions of photos printed on digital paper. All three of these interaction contexts, as is also true for Anderson *et al's* lecture study, are of human-human interactions in which participants share a public writing space.

2.1.2 Out-Of-Vocabulary Words In Natural Speech Processing Contexts

Yu *et al.* found that out-of-vocabulary (OOV) terms account for between 4.2%-8.0% of the unique words occurring in meetings [179]. These OOV words tend to be those that are important for understanding, like keywords, technical jargon, proper names [138] or abbreviations. In examining OOV words in French news broadcasts, Allauzen and Gauvain [3] reported that as much as 70% of OOV words were named-entities, like proper names. Palmer and Ostendorf [138] found that on a corpus of English news broadcasts 45.2% of OOV words occurred in phrases denoting a person, although proper names made up only 3% of all the words occurring in the corpus. For search queries (like those one might type into Google) spoken documents are represented by their speech recognition transcripts. When OOV proper names are mis-recognized in those transcripts then it has a detrimental impact on retrieval accuracy [101].

In a recent analysis of lecture speech, Glass [60] pointed out that the ideal vocabulary for speech recognition is not the largest vocabulary, but rather one that is both relatively small and has a small Out-Of-Vocabulary (OOV) rate. A small vocabulary minimizes substitution errors, and a small OOV rate minimizes insertion errors. The problem is that in general the size of vocabulary and the rate of OOV are inversely proportional. To illustrate the difficulty of obtaining such vocabularies, Glass compiled a small, 1.5K vocabulary of words common to lectures in three different course areas. Commonness was measured by how many lectures a word occurred in. The finding was that still in each lecture area the 10 most frequent subject-specific words were not common across the corpus of lectures, and thus OOV. The presence of technical, subject-specific OOV terms makes deriving a vocabulary and language model for lecture speech and other natural speech contexts a significant challenge.

When a lecture topic area or its technical vocabulary are known ahead of time, then automatic vocabulary expansion can be used [179, 128], which leverages textbooks or targeted web searches to augment dictionaries or language model statistics. Kurihara *et* *al.*, in their work on the use of predictive handwriting during lectures given in Japanese [99], assure full coverage with such methods. But such techniques don't work for less formal interactions, like a spontaneous whiteboard brainstorming session. Palmer and Ostendorf used various off-line sources of personal name information to reduce OOV names in broadcast news recognition by as much as 40% [138]. Their technique used an acoustic distance measure to compare phonetic sequences representing OOV names to the names in the off-line lists. Phonetic distance was based on a data-derived phonetic confusion matrix. Matches that were close enough triggered the addition of an appropriate off-line name to the system's on-line recognition vocabulary. Later, in Section 5.3.3, we'll discuss the differences between this approach and SHACER's approach to measuring phonetic distance, which compares handwriting to speech rather than text to text as in this work of Palmer and Ostendorf.

Work in the area of spoken document retrieval (SDR) [58] also must deal with OOV terms. Saraclar and Sproat [153], while performing speech recognition on a corpus of six teleconferences, using a vocabulary and language model from the Switchboard corpus, reported a 12% OOV rate. Using the same vocabulary and language model on Switchboard data itself had only a 6% OOV rate. As the OOV rate increased in moving from Switchboard to Teleconference data so too did the recognition word-error-rate (WER) with an attendant loss in precision-recall rates for document retrieval. Yu *et al.* [181], also using a vocabulary and language model from the Switchboard corpus, found a 39.7% OOV rate for keywords (e.g. *semiconductor*, *radio-shack-solution*, and *multiple-database-search*) occurring in voicemail segments from the LDC Voicemail corpus [137]. Palmer and Ostendorf [138], in examining English language news broadcasts, found that with a relatively large dictionary (57K words) and an OOV rate below 1%, still 10-15% of recognized broadcast news sentences had at least one OOV — and these OOVs were likely to be named entities that carried much of the sentence's meaning.

Although larger vocabularies lead to lower OOV rates in natural speech contexts, larger vocabularies are also more expensive computationally (e.g. require more pruning, etc.) and require more memory than smaller vocabularies [138]. Rosenfeld [146] found that increasing vocabulary size for recognition of read North American Business news text beyond an optimal 64K did not significantly improve recognition. Although increasing the system vocabulary size did help recognition rates for many common words, it actually hurt recognition rates for less common words by increasing the likelihood of insertion errors. Less common words, like proper names, typically carry more semantic information, so insertion errors due to increased vocabulary size don't solve the OOV problem. Introduced insertion errors are still detrimental to understanding.

2.1.3 Dynamic Learning Of Out-Of-Vocabulary Words

In the previous section it was made evident that simply increasing vocabulary size will not solve the OOV problem. There is a need for some more dynamic means of detecting and properly recognizing OOV words. SHACER's goal is to dynamically learn OOV terms, as they are presented redundantly during the course of an interaction. SHACER aims to dynamically learning the spelling, pronunciation and local semantics of new terms, enrolling them into dictionaries and language models as the system is running, and thus improving the system's accuracy and understanding over time and usage.

Dynamically learning new vocabulary has also been demonstrated by systems that cluster similar repeated phonetic sequences, refine those clusters and then associate the refined pronunciation representing the cluster to recurring images or actions [176, 150, 147, 175]. In the context of speech-only systems repetitive acoustic patterns can be identified for word discovery during lectures [141]. Explicit pronunciation learning for new name enrollment can be accomplished by having users first say and then spell their names [39].

2.1.4 Multimodality In Learning And Teaching

Moreno and Mayer's theory of multimedia learning [117] is founded on three working assumptions drawn from cognitive psychology [10]: (1) humans have separate processing systems for visual/pictorial versus auditory/verbal channels of information (dual-channel assumption), (2) each processing channel has limited capacity (limited-capacity assumption), and (3) that meaningful learning requires mental processing in both verbal and visual channels, building connections between them. These assumptions are discussed also in Wickens *et al.* [170].

Given these assumptions, Mayer and Moreno [109] can explain why presenting text that is also spoken helps students learn more effectively, while presenting visual animations or graphics along with visual and spoken text hurts learning. When the redundancy is across two channels (visual and auditory) then processing proceeds in parallel in both channels and the effect is complementary. When the redundancy is in the same channel (e.g. a visual graphic with accompanying visual text) then the focus of attention must be split overloading cognitive processing and resulting in degraded learning performance. The eyes physically have to move back and forth focusing first on the graphic and then on the text, and this can lead to cognitive overload. Wickens *et al.* in their *Four-Dimensional Multiple Resource Model* agree with the conclusions of Mayer and Moreno, that it, "...is apparent that we can sometimes divide attention between the eye and ear better than between two auditory channels or two visual channels."

The import of Mayer and Moreno's findings is that students have better recall and learn more effectively when textual information is presented redundantly in both visual and auditory modes. Next we will show that in some human-human interactions speakers typically present information in just this way, redundantly across both visual and auditory channels, by handwriting words and also saying them

2.2 STUDY OF MULTIMODAL REDUNDANCY

We collected data in three settings: (1) online whiteboard presentations (WP), (2) a spontaneous brainstorming (SB) session, and (3) photo annotation (PA) discussions. The methodology was to annotate all handwriting and speech. For the redundancy analysis, the frequency with which handwritten words were accompanied by redundant speech was examined.

2.2.1 Term Frequency - Inverse Document Frequency

Tf-idf word weights [1] are commonly used in search and retrieval tasks to determine how important a word is relative to a document [11]. Words that occur with high-frequency in a document, but are relatively rare across the set of documents under consideration, provide a good indication of the document's content [151]. For *tf-idf* analysis documents were constructed by concatenating the transcripts of both the spoken and handwritten words for each discourse segment of the collected data.

The handwritten abbreviations shown in Figure 2.3 (e.g., J, LB) exemplify the relation between dialogue-critical words and *tf-idf* weight. They are dialogue-critical words because without knowing how they are grounded in speech, as shown by the call-outs in Figure 2.3 (J = "Java tier," LB = "Load Balancer"), the underlying visual representation lacks meaning. They also have high *tf-idf* weights because they occur frequently within the presentation, but not so frequently across the entire set of presentations. Thus the abbreviations in Figure 2.3 are both dialogue-critical and highly weighted.

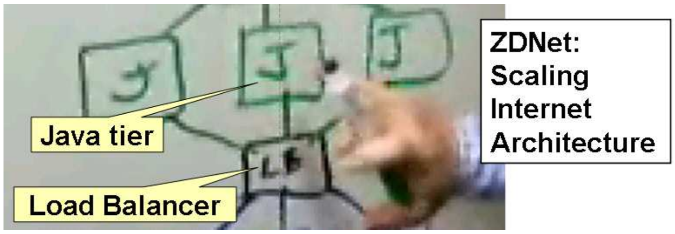


Figure 2.3: Dialogue-critical words are those whose grounding must be known in order to understand the presentation or discussion (e.g., J = "Java tier", LB = "Load Balancer")

[1] Wikipedia entry for *tf-idf*: There are many different formulas used to calculate *tfidf*. The term frequency (TF) is the number of times the word appears in a document divided by the number of total words in the document. If a document contains 100 total words and the word cow appears 3 times, then the term frequency of the word cow in the document is 0.03 (3/100). One way of calculating document frequency (DF) is to determine how many documents contain the word cow divided by the total number of documents in the collection. So if cow appears in 1,000 documents out of a total of 10,000,000 then the document frequency is 0.0001 (1000/10000000). The final tf-idf score is then calculated by dividing the term frequency by the document frequency. For our example, the tf-idf score for cow in the collection would be 300 (0.03/0.0001). Alternatives to this formula are to take the log of the document frequency.

2.2.2 Corpora Description

Online Whiteboard Presentations (WP)

We examined 34 short (3-4 minutes) whiteboard presentations offered on ZDNet's *At The Whiteboard* site [182]. Figure 1.5 shows a partial frame from one of these presentations. These presentations discuss various technical and business topics (e.g. "Pain-Free Annual Budgeting," "The B2B Marketing Challenge," etc.). There were on average of 11.6 handwriting events per presentation, and within those events 15.9 annotatable handwritten words. In the 34 presentations there are 33 different presenters. The presentation videos are professionally made, and the speakers are in general practiced. Half of the presenters were associated with ZDNet, and half were executives from other companies (e.g. Dell, Intel). Twenty nine of the presenters were male, and four were female.

Audio and video annotations were done by hand using WaveSurfer's [167] video transcription plug-in. Handwriting was annotated by scrolling the video frame-by-frame (video was recorded at 15 frames per second) to mark the moment of initial pen-down and final pen-up for each handwriting instance. If only one of the pen-up/pen-down events could be clearly seen then the annotator made a best estimate for the other if possible, and if not possible or if neither event could be clearly seen then the handwriting instance was not counted.

Second Scoring: Handwriting Annotation Reliability

In order to judge the reliability of our annotation protocol, a second annotator scored five randomly selected presentations from among the thirty-four, i.e., a 15% random sample. Compared to the first annotator there was a 100% match on what the annotatable handwriting events were, a 96% match on the handwritten words within each event, and a 99% match on the spelling of matched words. The *kappa coefficient* [2] was 0.92 for agreement on which instances of handwriting were actual annotatable words. For the five second-scored presentations there was a total of 84 handwritten words: 27 were un-annotatable (32%

[2] *Kappa coefficient*: a standard measure of inter-rater reliability [34]. Scores $>$ 0.8 indicate good reliability.

of total), and 57 were annotatable (68% of total). Between annotators the handwriting instance start times varied on average by 71 milliseconds and the end times by 49 milliseconds. Rounding up, the handwriting annotation timing accuracy was reliable to within 0.1 seconds, which is the same standard of reliability used in previously published studies on multimodal timing by Oviatt *et al.* [130].

Spontaneous Brainstorming Session (SB)

Multimodal redundancy also occurs in less formal situations. For example, we recorded a spontaneous brainstorming session, which occurred during a two day planning meeting with twenty participants. Ninety minutes of the session were recorded. Figure 2.4 is an example of handwriting and speech that occurred during this session. Annotation of handwriting events followed the same procedure used in annotation of the ZDNet whiteboard meetings (see above). For audio transcription, only speech that was associated with a handwriting event was annotated.

All handwriting was performed by the session leader, but the speech associated with the handwriting events was spoken by various participants in the meeting. Only 52% of the speech accompanying the presenter's public handwriting during the brainstorming session was spoken by the handwriter. The other 48% was spoken by seven out of the other twenty meeting participants. The percent of contributions from each of those seven roughly matched their positions in the organizational hierarchy underlying the meeting. So, the project manager's contributions were greatest (14%) followed by those of the project lead (9%), team leads (9%, 5%, 5%) and then of the project engineers (5%, 3%).

Terminology

In a document, each unique word is referred to as a word *type*, while each individual word occurrence is referred to as a word *token*. If while saying "hand over hand" a presenter also wrote the word *hand*, then concatenating the speech and handwriting transcripts would yield the word token list, "hand over hand hand," with three tokens of the word type, *hand*. We refer to the word types in this combined token list as *overall* types (i.e., *ALL*) because they can originate from either speech or handwriting. The subset of *ALL* word

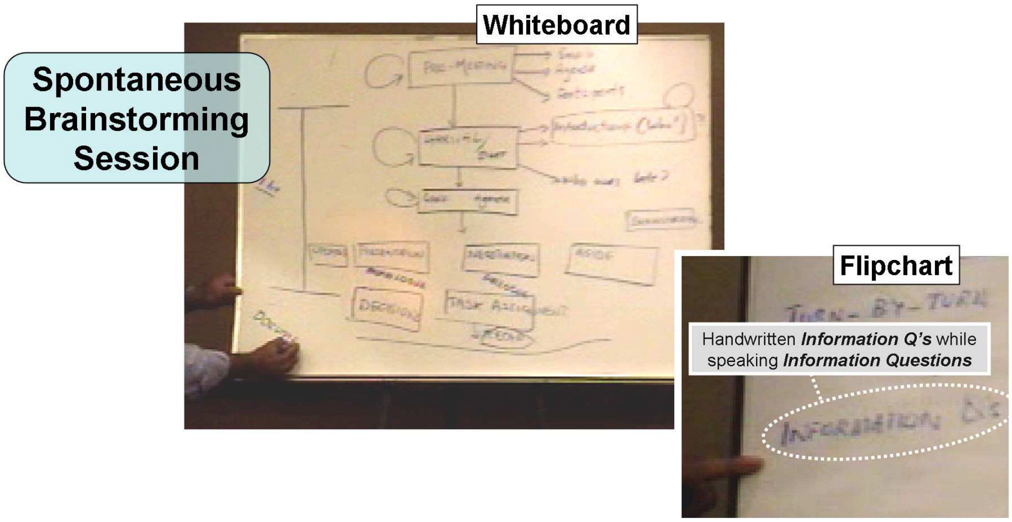

Whiteboard
Spontaneous Brainstorming Session
Flipchart
Handwritten Information Q's while speaking Information Questions

Figure 2.4: The whiteboard and flipchart of a ninety minute spontaneous brainstorming session. Virtually all handwriting (98%) was also spoken redundantly by various participants in the meeting.

types that were handwritten are HW word types. The subset of HW types that were redundantly handwritten and spoken are RH types.

In natural language processing tasks, a stop-list typically contains closed class words like articles, prepositions, pronouns, etc., which tend to occur with equal relative frequency in most documents. When computing *tf-idf* weights [11], the stop words (i.e., words occurring on the stop-list) are removed from consideration, because they tend to add little to the determination of which words are important representatives of a particular document.

Photo Annotation (PA) using Digital Paper and Pen

In [17] we reported on some aspects of a pilot study in which photos printed on digital paper were discussed and simultaneously annotated with a digital pen (as shown in Figure 2.1 and Figure 2.5). See [17] for a discussion digital pen/paper technology. There were four annotation sessions. In this thesis we further analyze data from the two native English speakers' sessions. All speech for these photo annotation sessions was hand annotated, but the handwriting gestures were automatically captured via digital paper and pen. Figure 2.5 shows three participants discussing a photo, using a digital pen to annotate the photo image printed on digital paper. The photo under discussion is shown at the bottom of Figure 2.5. An image of the digital paper was projected, so they could be seen more easily. All annotations written on the digital paper were rendered on the projected image as they were created.

Participants were asked to choose some photos they'd like to discuss (9 and 10 photos each for the sessions reported on here). They then spoke about their photos to a small group of others, having been told that they could annotate freely and that the software would process their annotations so they would get back labeled photos. Photos were automatically projected on a shared display, since not all discussion members could easily see the paper versions. The projected images were automatically updated when touching the digital pen to a photo sheet (*cf* [17]).

2.2.3 Study Results

Percent of Handwriting Time

Previously, Kurihara *et al.* found that as much as 18% of lecture time was spent handwriting [99]. For the ZDNet whiteboard presentations examined here, the presenters spoke on average for 192.9 seconds (stddev = 44.3 seconds) and handwrote on average for 38.9 seconds (stddev = 20.9 seconds). Thus, on average 21.3% (stddev = 13.4%) of presentation time was spent in handwriting.

For 8 of the 34 presentations we have sketch annotations also. On average for those eight presentations, sketching occurred during 24.8% of the presentation time (± 0.04%), with presentation time measured from start-of-speaking to end-of-speaking. Thus it appears that on average there was a slightly larger time-wise-percentage of sketching than handwriting during these presentations.

Redundancy

Table 2.1 shows the number of handwritten words that occurred in each of the three corpora (*HW* row), along with the number of handwritten words that were also spoken redundantly (*RH* row). The bottom row of Table 2.1 shows the percent of handwritten words that were spoken redundantly (*RH*/*HW* row). The total number of handwritten

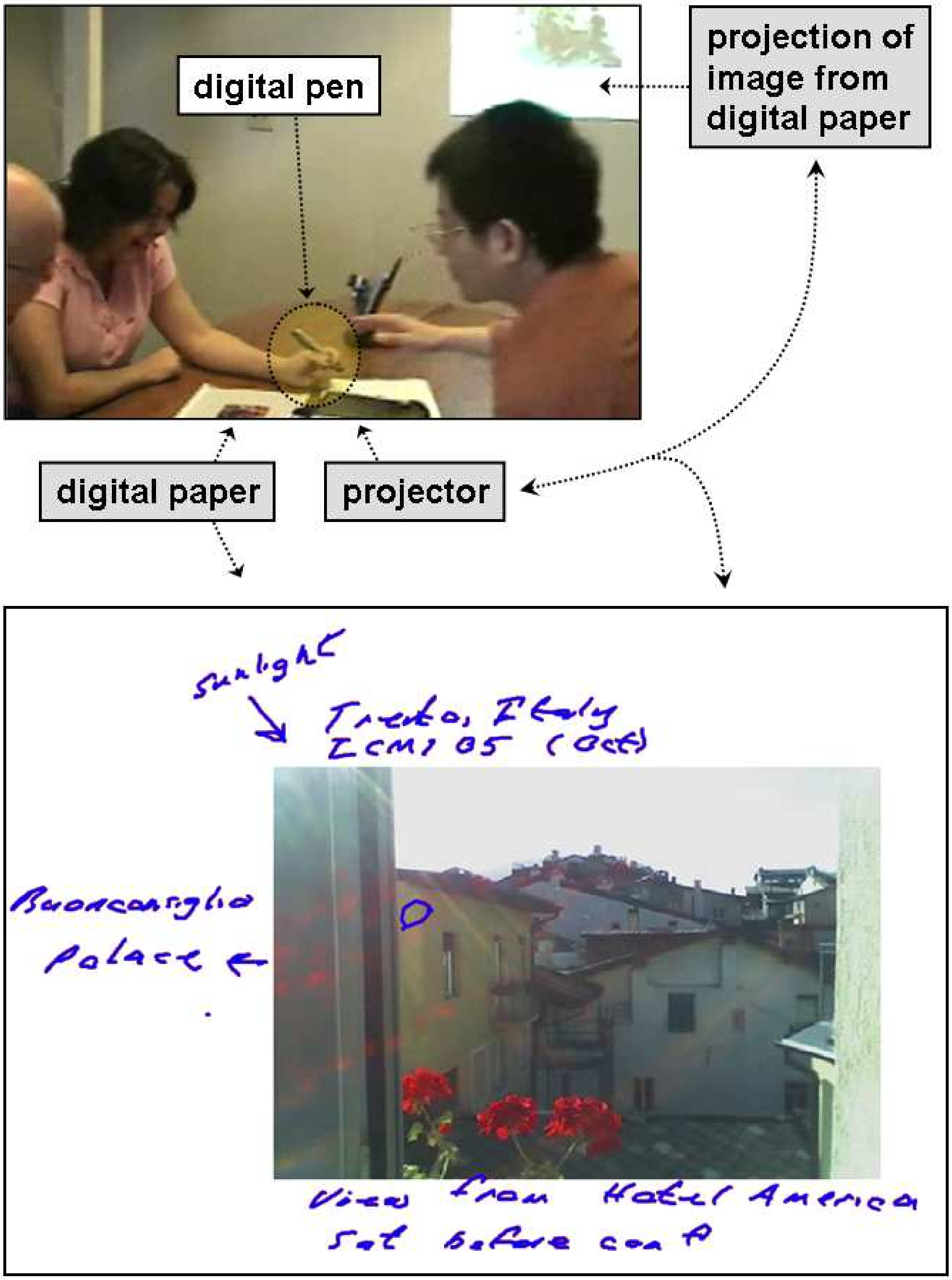
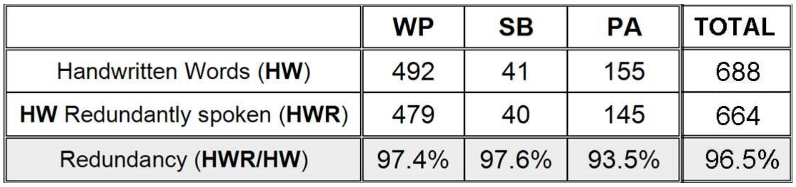


Figure 2.5: Three photo annotation session participants discuss a photo printed on digital paper. As the narrator handwrites annotations they are also projected so the cross-table participant can view them more easily.

words accompanied by redundant speech (TOT column in Table 2.1) over all three corpora was 664 out of 688 words, for an overall redundancy rate of 96.5%. These results support the claim, which is derived from our working hypothesis, that multimodal redundancy is typical of human-human interaction settings where multiple modes can be perceived

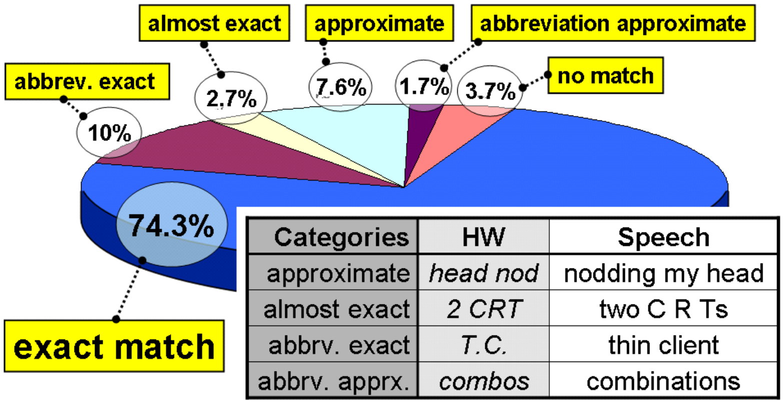
Table 2.1: Word-level redundancy rates across ZDNet whiteboard presentations (WP), the spontaneous brainstorming (SP) session, and the photo annotation (PA) discussions.



	WP	SB	PA	TOTAL
Handwritten Words (**HW**)	492	41	155	688
HW Redundantly spoken (**HWR**)	479	40	145	664
Redundancy (**HWR/HW**)	97.4%	97.6%	93.5%	96.5%

Figure 2.6 shows the types of redundant matches that occurred averaged over all three corpora. The preponderance of matches are exact lexical matches (74.3%), where the handwritten terms are spoken exactly as written. Abbreviation exact matches are defined as standard abbreviations that exactly match their expansions in speech (10% — see Figure 2.4 inset). Almost exact matches differ only in number or tense (2.7%). Approximate matches differ in word order or form (see table inset in Figure 2.6), or have extra or missing words (7.6%), as is also true for the spoken expansions of abbreviation approximate matches (1.7%). Category examples are shown in the inset of Figure 2.6. Our result of 74.3% exact match with 96.5% overall redundancy closely parallels the 74% exact match and 100% redundancy found earlier by Anderson *et al.* [5]. However, Anderson *et al.* examined only 54 instances of handwriting. This work has analyzed an order of magnitude more data — 688 handwriting instances. Our findings are thus numerically more significant. We also examine three different scenarios, none of which was based on the use of a tablet PC as in the study by Anderson *et al.*

For the ZDNet corpus, the percentage of handwritten words that were part of a name-phrase was 7.3% (e.g. *Sarbanes Oxley*, *Ghostbuster*, *Lawson*), while for the two photo annotation sessions that average percentage was much higher at 46.4% (e.g. *Jenolan Caves*, *Kayla*, *Buonconsiglio Palace*). Recall that the percent of words in named entities for French news broadcasts was about 1.2% [3] and for English broadcast news it was about



Categories	HW	Speech
approximate	*head nod*	nodding my head
almost exact	*2 CRT*	two C R Ts
abbrv. exact	*T.C.*	thin client
abbrv. apprx.	*combos*	combinations

Figure 2.6: Redundancy category breakdown averaged across ZDNet whiteboard presentations (WP), the spontaneous brainstorming (SP) and photo annotation (PA) sessions. Category examples are shown in the inset table.

3% [138]. The rate of words that occurred in handwritten name-phrases for presentations and photo annotation discussions was three to forty times higher than occurrence rates for broadcast news. Miller *et al.* [113] examined the OOV rate of various named-entity types for various vocabulary sizes across English news broadcasts. Their data for proper names are given here as vocabulary-size/OOV-rate pairs: 5K/65%, 10K/47%, 20K/30%, 40K/15%, 60K/11%, 80K/9%, 120K/6%. These OOV rates were as much as a factor of ten greater than the baseline OOV rates for non-name words. Miller *et al.* [113] also state that increasing vocabulary size above 60K does not improve recognition, because it introduces more errors than it fixes. Thus, for the presentations and photo annotation sessions examined here handwritten name-phrase words occurred much more frequently than in news broadcasts, and are therefore more likely to be OOV in these natural speech contexts.

The percentage of handwritten name-phrase words for the ZDNet presentations was much less than for the photo annotation sessions; however, the ZDNet presentations had many more abbreviations than the photo annotation sessions. In the ZDNet corpus, 44.3% of handwritten words were abbreviations, while for the two photo annotation sessions only 5.7% of the handwritten words were abbreviations. Presentation-specific abbreviations may be missing from standard recognition vocabularies and thus are also likely to be OOV. An example of a presentation specific OOV abbreviation was given in Section 1.2 where *CAGR* was spoken as *Category Growth Rate* in contrast to its standard interpretation of *Compound Annual Growth Rate.* Non-standard abbreviations also occur in biomedical articles. Yu *et al.* developed a tool for mapping abbreviations in biomedical articles to their full forms [178], but after applying their expansion tool there were still 75% of abbreviations remaining undefined. Of these undefined abbreviations, fully 32% could not be found in any of four standard medical abbreviation databases. Thus non-standard/idiosyncratic abbreviations present difficult problems for recognition and understanding just as is the case for OOV named entities.

Of the six different types of redundancy charted in Figure 2.6, SHACER can currently take advantage of three — (1) exact, (2) abbreviation exact, and (3) almost exact redundancies. These three categories represent 87% of the handwriting events reported on in this thesis. Within the no match category (3.7%) there was a sub-category dubbed semantic matches. These are cases in which, for example, a narrator while writing the name of a family member (e.g. Donald) says both the relationship and name of that family member (e.g., "my son, Donald"), and then later while again writing the name says only the relationship, "my son.". Such semantic matches occurred in about 1% of redundant instances. In the future SHACER may be able to benefit from such semantic matches as well as from the broader category of approximate matches.

Redundancy Timing

Understanding the temporal relationship between handwriting events and redundant speech is important. If they are likely to be temporally close then the search space for aligning and detecting such redundancies can be reduced.

Following Oviatt *et al.* [136] we examined the integration patterns of redundantly delivered inputs. For the 34 ZDNet presentations we examined the sub-corpus of the 382 handwriting/speech matched input instances (out of 395 total instances). Note that these 395 handwriting instances contained the 492 handwritten words listed in Table 2.1's *WP* column as having occurred during the 34 ZDNet Whiteboard Presentations — thus

Table 2.2: Temporal categories by precedence (for ZDNet corpus). Note that 24% of instances are sequential (left), with no overlap between handwriting (W) and speech (S).

Sequential		Simultaneous (Over-lapped)		
Writing First	Speech First	Speech Precedes	Writing Precedes	Neither Precedes
(16%) W__ S__	(8%) S__ W___	S_____ W___ (1%)	S___ W_____ (6%)	S_____ W_____ (0%)
		S_____ W__ (11%)	S__ W_____ (39%)	S____ W__ (2%)
		S___ W____ (1%)	S____ W___ (15%)	S___ W_____ (1%)

some instances included more then one word. We found that 24% of inputs were presented sequentially with either handwriting occurring first followed by speech (Table 2.2, *Writing First* — 16%), or speech occurring first (8%). For simultaneous (over-lapped) constructions (76% of instances), speech preceded handwriting in 13% of cases, handwriting preceded speech in 60% of cases, and neither preceded in 3% of cases (timing accurate to 0.1 sec). Thus, overall, for the majority of redundant handwriting and speech instances (60%) the inputs are overlapped with handwriting preceding speech. As in Oviatt's study the tendency of handwriting to precede speech was significant by Wilcoxon signed ranks test, T+=524.5 (N=32), p<.0001, one-tailed.

When we superimpose the timing data for all instances (both sequential and simultaneous) from the spontaneous brainstorming session onto to that of the ZDNet presentations (Figure 2.7), the timing contours are closely matched for the session leader while diverging somewhat for the brainstorming session's other participants. Figure 2.7 takes all of the redundant instances and categorizes them into time-delay bins. The percentage of redundancies in each bin, grouped by seconds from start-of-handwriting to the start-of-speech, is plotted in Figure 2.7. Negative values mean that speech occurred first. During the spontaneous brainstorming session when handwriting was spoken redundantly by others (rather than by the leader), there was a marked shift in the peak average amount of time by which speech trailed handwriting. Thus when speaking about his own handwriting the brainstorming session leader's timing pattern closely matched that of the average ZDNet presenter — with handwriting slightly preceding speech and simultaneously overlapping it. However, when the speech of other meeting participants was reflected in his handwriting, he was first listening to what others said, extracting key terms, and then after a few seconds (i.e., 4-6 seconds) handwriting those key terms on the whiteboard or flipchart.

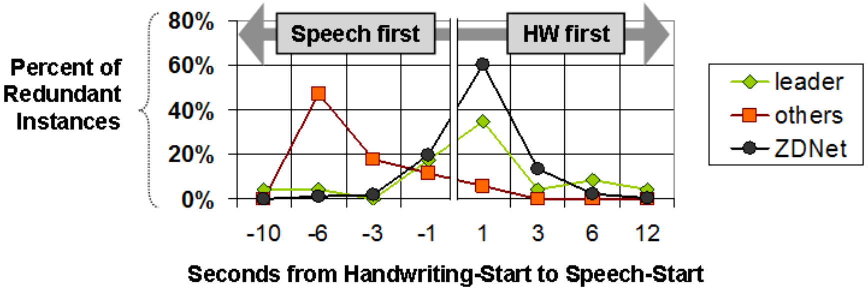


Figure 2.7: The number of seconds by which the start of handwriting (HW) preceded the start of speech. Negative values mean that speech preceded HW. The plot includes data from the ZDNet presentations (WP) and from both the brainstorming (SP) session's leader and other participants. Note that, as expected, redundant speech not from the leader occurred much later relative to handwriting than the speech of the handwriter.

Of the sequential inputs shown in Table 2.2, 33% were speech followed by handwriting, a pattern which for speech and sketched graphics in Oviatt *et al.* [136] occurred for only 1% of the sequential inputs. This much larger proportion of speech preceding handwriting (33%) versus Oviatt *et al.*'s small proportion of speech preceding sketch (1%) may reflect some qualitative difference between handwriting and sketching. Perhaps handwriting requires more cognitive effort and is therefore delayed in presentation compared to locative sketching.

Inter-modal lag times are the amount of time that elapses from the end of the first sequential mode to the start of the next mode. The inter-modal lag times for the sequential patterns (shown in Table 2.2) are charted in Figure 2.8. In both cases (i.e., the speech first case, and the handwriting first case) most of the lags were less then 2 seconds. This can be seen by adding together the two left-most column groups in Figure 2.8): 80% of the speech-first lags and 76% of the handwriting-first lags are less than 2 seconds. For the speech first condition all lags were within 4 seconds. For the handwriting first condition 8% of the lags were longer than 4 seconds, with the longest being a full minute and a half, by one of the ZDNet presenters. This temporal analysis means that for natural speech settings like those analyzed here, redundant speech can usually be found temporally close to the handwriting. When handwriting occurs it is likely to be accompanied by redundant speech. In 90% of the cases that redundant speech will most likely occur within 4 seconds before or after the start of the handwriting instance. These lag-times are quite close to the temporal results for bimodal speech and sketch inputs studied by Oviatt *et al.* [136] [3]; however, in their study 99% of sequential constructions were pen-first inputs whereas in our study only 67% of sequential inputs were handwriting-first.

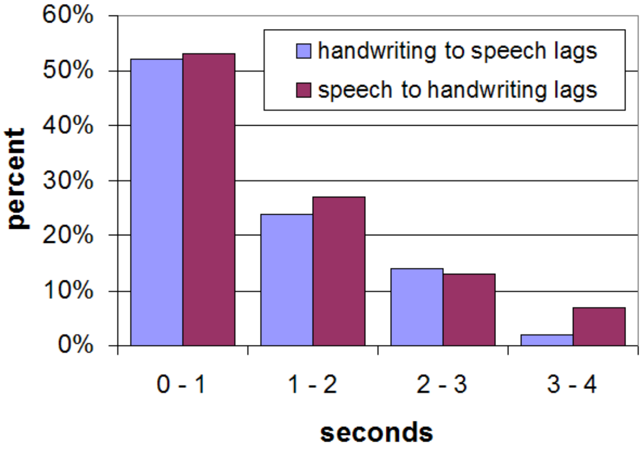


Figure 2.8: Modal lag between sequential handwriting and speech (or vice versa). Most lag times fall below 4 seconds.

Redundancy as a Structural Marker

Table 2.2 does not include data from the brainstorming session, but there too some long lag times occurred between handwriting and speech. Such temporally distant redundancy seemed to serve a somewhat auxiliary function in not focusing attention immediately but rather in introducing the expectation of that focusing event in the future, and thus serving as an indicator of meeting structure. For example, at one point in the brainstorming session

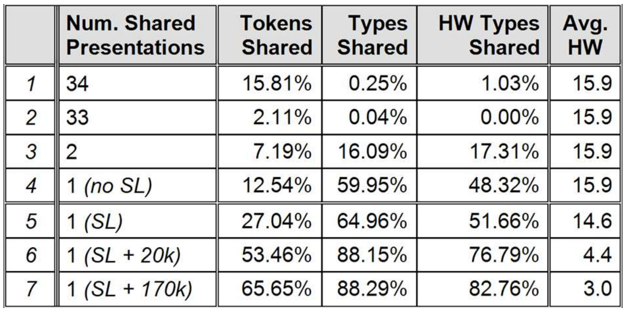
[3] Oviatt *et al.* [136]: “The lag between the end of the pen signal and start of speech averaged 1.4 seconds, with 70% of all lags ranging between 0.0 and 2.0 sec, 88% between 0.0 and 3.0 sec, and 100% between 0.0 and 4.0 sec.”

the presenter wrote a discussion point on his flip-chart, but before saying it relinquished the floor to another speaker. Several minutes later, at an appropriate juncture, the presenter took the floor back by pointing at and verbally referring to his previously written place-holding discussion point. In the next several dialogue turns he and others spoke the place-holding term five more times with accompanying redundant deictic pointing. This redundantly introduced term was both dialogue-critical (as evidenced by the number of times it was repeated) and also served as a gloss to meeting structure by marking the expectation of an important topic of interaction.

Redundancy and Projected Out-of-Vocabulary Words

Glass *et al.*, in [60], examined the nature of OOV words in a small general vocabulary of words common to a training set of lectures (see Section 2.1.2), and found that subject-specific words from lectures were not well covered and often missing even from the vocabulary of larger corpora like Broadcast News (recorded and transcribed television and radio broadcasts) and Switchboard (spontaneous, two-party telephone conversations on 50 different topics). Here we perform a similar examination of word-type sharing across the 34 presentations of our ZDNet whiteboard presentation corpus.

Table 2.3: Percent of tokens and types common to a Number of Shared Presentations. Percent of handwritten (HW) types commonly shared is also given, as well as the average number of HW types ($Avg.\ HW$) per presentation.



	Num. Shared Presentations	Tokens Shared	Types Shared	HW Types Shared	Avg. HW
1	34	15.81%	0.25%	1.03%	15.9
2	33	2.11%	0.04%	0.00%	15.9
3	2	7.19%	16.09%	17.31%	15.9
4	1 *(no SL)*	12.54%	59.95%	48.32%	15.9
5	1 *(SL)*	27.04%	64.96%	51.66%	14.6
6	1 *(SL + 20k)*	53.46%	88.15%	76.79%	4.4
7	1 *(SL + 170k)*	65.65%	88.29%	82.76%	3.0

Table 2.3 shows the results of examining the number of shared word tokens and word types along with the number of shared handwriting (HW) types (type = unique word, token = an individual occurrence of a word type). Row 1 of Table 2.3 shows that across

all 34 presentations 15.81% of word tokens were shared, while only 0.25% of word types and just 1.03% of handwritten types were shared commonly. This means that a relatively large number of tokens for a small number of types were shared. This is common when there is no stop word removal, because for example all the presentations share many occurrences of articles like "a," and "the." With no stop list removal the average number of handwriting types per presentation was 15.9. There were 209 average overall word types per presentation. The percent of overall word types occurring in only one presentation was 59.95% and of handwritten types was 48.32% (Table 2.3, row 4 — no stop-list removal (*no SL*), *Types Shared* and *HW Types Shared* columns). Row 2 of Table 2.3 shows that it is rare for tokens or types to occur in all presentations but one. Row 3 of Table 2.3 shows that it is also relatively rare for tokens and types to occur in only two presentations and no others.

In the lower three rows of Table 2.3 (rows 5-7) we show that the percentage of shared types increases after basic stop list removal (*SL*, row 5) and then further increases with larger dictionaries combined with the basic stop list removal (*SL + 20k* and *SL + 170k*, rows 6-7). Note that *SL* = basic stop list, 20k = a 20,000 word dictionary of the most common words in a corpus of meetings, and 170k = a 170,000 word dictionary from the Festival Speech Synthesis Toolkit [21]. It can be seen that as the number of words removed by combined stop-list and dictionary removal increases the number of remaining presentation-specific handwritten types decreases from 14.6 in row 5 to only 3 in row 7. Thus with a large general dictionary (e.g. 170k) the roughly 8 presentation-specific handwritten types present in row 4 ($48.32\% \star 15.9 \approx 8$) are reduced to just 2 in row 7 ($82.76\% \star 3 \approx 2$). However, employing such a large general vocabulary (e.g. 170k) to reduce potential OOV words is not ideal, as Glass and Palmer have pointed out [60, 138], Larger dictionaries require more computational resources and are susceptible to higher word-error rates due to substitution errors.

In order to avoid reverting to oversized dictionaries, an important question is then, if we had many more presentations to examine could we hope to find a smaller dictionary with better OOV coverage? Figure 2.9 shows a power regression prediction that addresses this question. As in row 4 of Table 2.3, the plotted points in Figure 2.9's left-side plot are computed with no stop list removal. To accumulate the data points plotted in Figure 2.9's left-side, we processed successively larger subsets of our corpora (i.e., 1 meeting out of the 34, 2 meetings out of the 34, 3 meetings out of the 34, etc.), asking for each increasingly larger subset how many overall and handwritten types occurred in only one presentation. The plot in Figure 2.9's left-side shows that the percent of presentation-specific overall word types (upper trend line) and redundant handwritten types (lower trend line) decreases steadily as set size increases, but at a rate that appears to level off. The power regressions were done in MS Excel $^{\mathrm{TM}}$, and the highlighted R-squared values indicate goodness of fit: 0.95 for overall and 0.97 for handwritten types (equations shown in Figure 2.9, left-side).

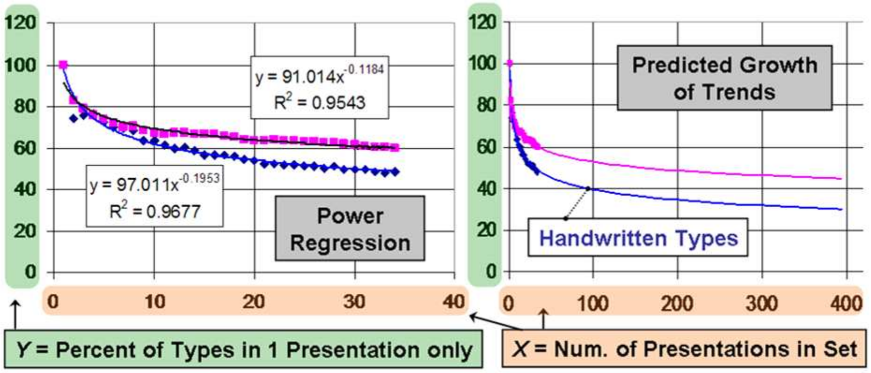


Figure 2.9: A power regression prediction to examine the percentage of handwritten word types occurring in only one presentation, given an increasing number of presentations. Upper line = overall types, Lower line = handwritten types.

In Figure 2.9's right-hand side plot we have extended the power regressions from the left-hand side to see what rate of presentation-specific handwritten words might still be present after examining a training set 10 times the size of our corpus. Trend lines were computed using the equations listed in the left-side plot of Figure 2.9. Trend lines were extended to 360 presentations. Even with this simulated order of magnitude larger training set there was still a relatively large percentage of presentation-specific handwritten types predicted (~30%, Figure 2.9, right-side, lower trend line). This finding indicates that in constructing an ideal vocabulary for a domain that includes public handwriting (e.g. lectures or meetings) as much as 30% of redundantly presented words are likely to be presentation-specific and thus out-of-vocabulary. This is in accord with the empirical evidence for proper name OOV rates at various vocabulary sizes given in Miller *et al.* [113] and paraphrased above. Next we will show that redundant handwritten words, which are likely to be highly presentation-specific, are indeed the dialogue-critical words that one would want to recognize and understand for later retrieval tasks.

Redundancy, *TF-IDF* Weight and Retrieval Searching

In earlier work [17] we showed that for photo annotation sessions, redundantly introduced (*RH*) words had a 90% higher average frequency than *ALL* word types. In this thesis we calculate the average *tf-idf* weights of *ALL* word types versus redundant handwritten (*RH*) word types, for not only the two native English-speakers' photo annotation sessions but also for the ZDNet corpus. For this combined data set, Figure 2.10 shows the average *tf-idf* weight increase for *RH* types compared to *ALL* types. These are strikingly higher *tf-idf* weights for *RH* types: 128% higher with no stop-word removal, and 70.5% higher with stop-word removal. These weight increases were significant by Wilcoxon signed ranks test (T+=561, N=33, p<0.0001, one-tailed).

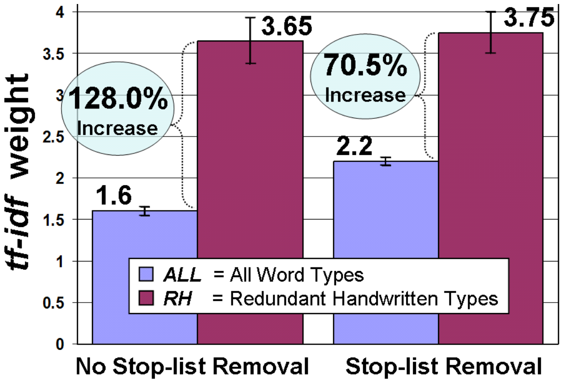
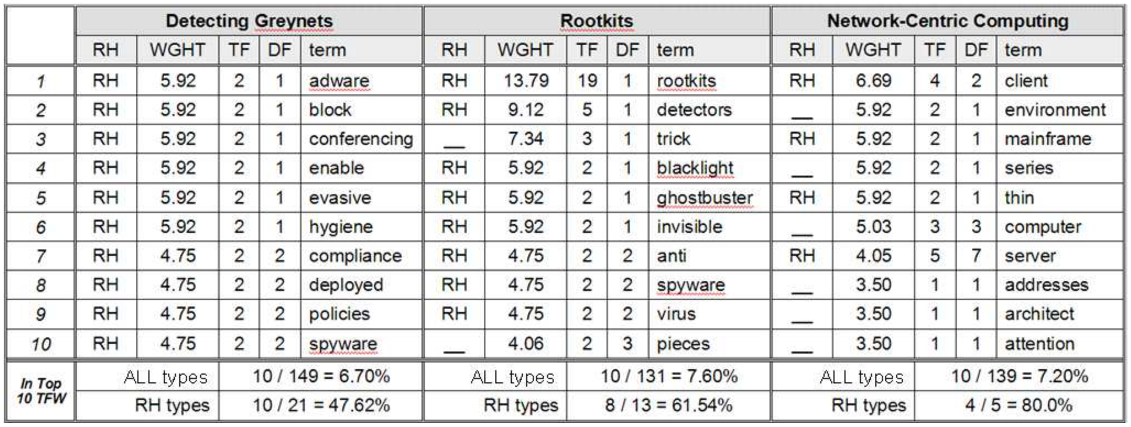


Figure 2.10: Average *tf-idf* weight increases for redundant handwritten word types (*RH*) versus *ALL* word types.

Table 2.4 shows examples from three ZDNet presentations of the top ten most highly *tf-idf*-weighted word types (after basic stop list removal). In some presentations — like the left-most, Detecting Greynets — all of the top ten are redundantly presented words. Some presentations had relatively lower percentages of redundant handwritten (*RH*) words in the top ten — as for the right-most, Network-Centric Computing. Even for these the RH words as a class were much more likely to be representative terms than non-RH words as a class (bottom rows, Table 2.4, *In Top 10 TFW*). On average for all 34 meetings only 7.66% of ALL types are present in the top ten most highly weighted words for a presentation. But of the redundant handwritten (RH) types, 61.47% are present in the top 10, which represents 48.64% of all top ten words for all presentations. Thus, the likelihood of a word being one of the top 10 most highly weighted words is less than 1 in 10 (7.66%) for ALL word types, while for RH word types it is about 5 in 10 (48.64%). This means that RH words as a class are significantly more representative of a presentation than non-RH words (Wilcoxon signed ranks test, T+=593 (N=33), p<0.0001, one-tailed). Similarly, on average, for all 19 individual photo discussions, just 11.5% of ALL types are present in the top 10 most highly weighted words. But of the RH types, fully 81.77% were ranked in the top 10, which represents 48.95% of all top ten words for all photo discussions.

Table 2.4: Top 10 word types ranked by *tf-idf* weight (WGHT) for three presentations from the ZDNet corpus. Key: RH = Handwritten and Redundantly spoken, TF = Term Frequency, DF = Document Frequency, TFW = *TF-idf* Weights.



	Detecting Greynets					Rootkits					Network-Centric Computing				
	RH	WGHT	TF	DF	term	RH	WGHT	TF	DF	term	RH	WGHT	TF	DF	term
1	RH	5.92	2	1	adware	RH	13.79	19	1	rootkits	RH	6.69	4	2	client
2	RH	5.92	2	1	block	RH	9.12	5	1	detectors	__	5.92	2	1	environment
3	RH	5.92	2	1	conferencing	__	7.34	3	1	trick	RH	5.92	2	1	mainframe
4	RH	5.92	2	1	enable	RH	5.92	2	1	blacklight	__	5.92	2	1	series
5	RH	5.92	2	1	evasive	RH	5.92	2	1	ghostbuster	RH	5.92	2	1	thin
6	RH	5.92	2	1	hygiene	RH	5.92	2	1	invisible	__	5.03	3	3	computer
7	RH	4.75	2	2	compliance	RH	4.75	2	2	anti	RH	4.05	5	7	server
8	RH	4.75	2	2	deployed	RH	4.75	2	2	spyware	__	3.50	1	1	addresses
9	RH	4.75	2	2	policies	RH	4.75	2	2	virus	__	3.50	1	1	architect
10	RH	4.75	2	2	spyware	__	4.06	2	3	pieces	__	3.50	1	1	attention
In Top 10 TFW	ALL types		10 / 149 = 6.70%			ALL types		10 / 131 = 7.60%			ALL types		10 / 139 = 7.20%		
	RH types		10 / 21 = 47.62%			RH types		8 / 13 = 61.54%			RH types		4 / 5 = 80.0%		

Table 2.4 shows that redundantly handwritten and spoken word types (RH) as a class are better representatives of their respective presentations or discussions than other words. Since they have significantly higher *tf-idf* weights than other words, they should be effective search query terms. To test this claim we performed retrieval queries on an indexed directory of speech and handwriting transcript documents, one such document for

each presentation in the ZDNet corpus. Documents for this *tf-idf* analysis were formed by concatenating the word-level transcripts of both speech and handwriting inputs. Each document represented one session, with a session being either one ZDnet presentation or the discussion surrounding one particular photo. The search engine we used was a state-of-the-art, open-source search application called Seekafile [157], which works on both small and large data sets. The point of this test was to confirm that using a *tf-idf* weighting scheme, which is typically applied to huge data sets, would be effective for searching over a small data set.

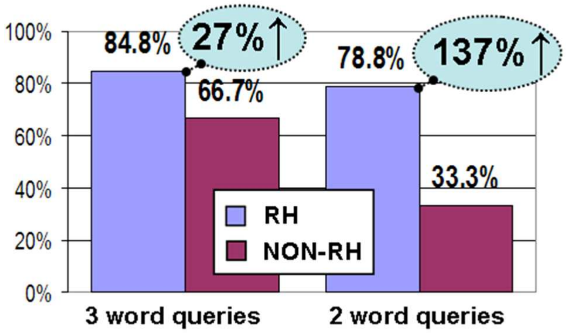


Figure 2.11: Retrieval accuracy using randomly selected three and two word queries, with words being chosen from the sets of redundantly presented handwritten word types (RH) and non-redundantly presented word types (non-RH).

We performed searches with both three-word and two-word queries (Figure 2.11). For each presentation the query words were randomly chosen from either the set of redundantly handwritten and spoken words (RH bars in Figure 2.11) or from the set of words that were not redundantly presented (non-RH bars in Figure 2.11). Retrieval accuracy measured how often the best-scoring retrieval result was the correct result.

The outcome for three word queries (Figure 2.11, left side) shows that words from the RH set yielded 84.8% retrieval accuracy while non-RH words yielded 66.7% accuracy. Thus for randomly chosen three word queries the retrieval accuracy was 27% higher using RH rather than non-RH words (marginally significant by Wilcoxon signed ranks test, $p<0.0655$).

For two word queries the right side bar chart in Figure 2.11 shows that randomly chosen words from the RH set yielded 137% higher accuracy than randomly chosen words from the non-RH set. RH accuracy was 78.8%, while non-RH accuracy was only 33.3%. Thus for two-word queries the retrieval accuracy was significantly higher using RH as opposed to non-RH words (Wilcoxon signed ranks test, T-=246, N=23, p<0.0001).

We have shown that as a class RH words have significantly higher *tf-idf* weights than non-redundantly presented (non-RH) words. Retrieval accuracy using RH words is significantly better than for using non-RH words. These results support the claim that redundantly presented words are more effective search query terms. They are thus measurably more important words in natural speech settings likes presentations or discussions than other words.

2.3 DISCUSSION

2.3.1 Study Implications

From the work of Moreno and Mayer [117, 109] on multimedia learning we know that redundantly presented words are easier to recall. In fact Glaser [59] states that adult learners remember only 10% of what they read, 20% of what they hear, 30% of what they see, but 50% of they see, hear and read — which is exactly the experience of a meeting participant in watching a presenter handwrite new vocabulary while also saying it. The multimodal convergence of handwriting and speech makes the event more memorable. This means that, after seeing redundantly presented words during a presentation, those words will later come to mind more readily for use in retrieval queries.

We have also shown that redundant words are likely to be presentation-specific. This means that for ideally sized vocabularies the percentage of redundant words likely to be OOV could be as much as 30%. We have also shown that redundant words in natural speech contexts like presentations and photo discussions have high rates of handwritten abbreviations and handwritten proper names respectively. Abbreviations and proper names as word classes are very likely to be OOV. Therefore Understanding these redundant OOV terms is critical for background understanding of a natural speech settings.

2.3.2 Multimodal Understanding of Human-Human Interaction

Shared public writing spaces are spaces that can be commonly viewed by a group of people participating in an interaction. Our claim is that terms handwritten in a public space during human-human interactions typically are also spoken. This claim has clear implications for the design of lecture or meeting transcription systems [60]. In other publicly viewed spaces, like those of television news broadcasts, multimodal redundancy also occurs in the form of textual labels that accompany on-screen images. If the high incidence of redundancy that occurs across handwriting and speech were similar for on-screen text and speech then multimedia data capture and indexing systems [163] could also hope to leverage it for increased understanding. Leveraging expected redundancies across note-taking and speech during call-center interactions could lead to better call-center data-mining applications [122]. Computational understanding systems deployed in all three of these settings could potentially benefit from leveraging multimodal redundancy across writing and speech.

2.4 SUMMARY

We have shown in this chapter that contrary to prevailing knowledge on modality usage in human-computer, multimodal command interface systems, handwriting and speech redundancy is not only a major organizational theme but is typical for some human-human interactions in which there is a public space for writing. Whereas Anderson *et al.* [5] in previously describing this phenomenon examined only 54 instances of handwriting in their study of computer mediated distance lecture deliveries, this chapter looked at an order of magnitude more data (688 handwriting instances) across three different contexts. We found that 96.5% of handwritten words were also spoken redundantly. These findings are both complementary to and numerically more significant than Anderson *et al's.*

Furthermore, we have shown that (1) a high proportion of redundantly presented handwritten words are likely to be out-of-vocabulary in relation to ideally sized recognition vocabularies, regardless of training set size, (2) that such redundancies are good mnemonic representatives of a presentation, and (3) that as a class they are significantly more representative of a presentation than other non-redundant word types.

Our working hypothesis was that people used multimodal redundancy to focus attention on important words. Derived from that hypothesis was the claim that if multimodal redundancy is a general communicative strategy, then it should be typical in human-human interaction settings. For the natural interaction contexts examined in this chapter we have shown that such redundancy is typical. The second claim derived from our working hypothesis was that if redundant words are dialogue-critical they should be measurably more important than other words. We have shown that redundantly presented words are significantly better retrieval query terms than non-redundant words. Therefore they are measurably more important. In the next chapter we will discuss SHACER, and show that these same important redundantly presented terms are dynamically learnable by unsupervised, boot-strapped methods.

Chapter 3

SHACER: Overview

Fixed vocabulary recognizers fail on out-of-vocabulary (OOV) terms. In the previous chapter we showed that in natural speech contexts both proper names and handwritten abbreviations occur frequently, are critical for understanding, and are also more likely than other words to be out-of-vocabulary. During presentations or meetings, when proper names or abbreviations are handwritten on a whiteboard or other public writing surface they are typically also spoken redundantly. It is not possible to expand vocabularies enough to cover all such OOV terms. Therefore there is a need to learn OOV terms dynamically. To address this need we have developed SHACER, a multimodal recognizer that leverages the occurrence of modal redundancy across handwriting and speech to discover the spelling, pronunciation and contextual semantics of OOV terms dynamically.

In this chapter we give a high-level overview of SHACER. First we briefly describe the task domain within which SHACER operates. Then we outline what SHACER is able to accomplish in that domain.

3.1 TASK DOMAIN

3.1.1 Gantt Charts

SHACER's task domain is the creation of whiteboard Gantt charts. Figure 3.1 is an illustration of a Gantt chart [1], drawn on whiteboard during a multi-party scheduling meeting. The horizontal axis is typically the time duration of the project, which may be

[1] A Gantt chart is a type of horizontal bar chart developed as a production control tool in the early 20th century by the American social scientist and engineer, Henry L. Gantt.

days, weeks, quarters or years. The project tasks are plotted as horizontal lines, whose start and end points indicate the approximate time period over which a task endures. Specific scheduling goals for a particular task are notated as diamond-shaped milestones placed on the task-lines.

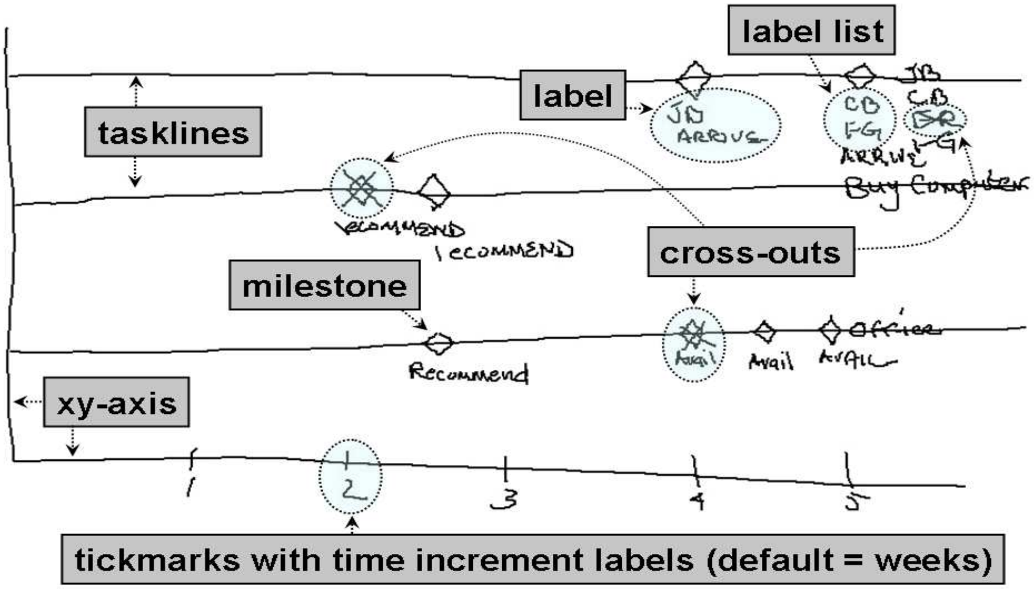


Figure 3.1: A typical Gantt chart from the SRI CALO Year 2 meeting corpus with callouts identifying the various recognizable constituent types.

3.1.2 Meeting Corpus

The corpus of meetings from which SHACER's inputs are drawn was collected at SRI International as part of the CALO project [89]. There are nine meeting series in the corpus, labeled *A-I*. Six of nine series are full five-meeting series. During collection of the earlier series (*A-F*) the collection hardware was being tuned, so their usefulness for development and testing of SHACER is limited. The *G* series of five meetings (*G1-G5*) is used as SHACER's development corpus and the *H* series (*H1-H5*) has been held out as the unseen test series.

Each of the meetings is around 5-10 minutes in length on average, and has three participants. The meeting participants were instructed to create a Gantt planning chart during each meeting using only the supported chart symbols: e.g. axes, lines, diamonds, tick-marks and cross-out marks (Figure 3.1). The story line of each meeting series involved hiring three new people, scheduling their arrivals, arranging for office space and equipment, dealing with delays, and choosing a printer for the new hires to share. This type of scenario-limited meeting is similar to those used in the European MultiModal Meeting Manager (M4) project [112], and the Augmented Multiparty Interaction (AMI) project [33]. In the SRI CALO meeting data there were five different sets of three people participating across the nine series of meetings. There was no dedicated prompter during the meetings as for the M4 collections, but the meetings' agendas and goals were set out in the pre-meeting instructions.

Participants were told how to label task-lines and milestones on the Gantt chart, and were asked to write out names in full while speaking them at first and then abbreviate them freely after that. This behavior presented no difficulties for the participants, and the speech and handwriting events in these meetings had timing and semantic relatedness properties similar to those seen in the lectures and brainstorming session studied in Chapter 2. Thus, although constrained, these meetings were realistic in the important aspects that made them suitable for the purposes of this thesis: (1) testing SHACER's ability to enroll new terms based on multimodal fusion of redundant events, (2) using the spelling, pronunciation and semantics of those new enrollments to semantically understand subsequent abbreviations, and (3) measuring the difference between system recognition rates using dynamic learning versus not using it.

3.1.3 Ambient Cumulative Interface

SHACER is deployed as part of a background perceptual application, which has been designed to observe human-human interactions, understand them, and produce useful background understanding. The interface is not a standard multimodal command interface; rather, it is an ambient cumulative interface or ACI for short. An ACI is a newly introduced class of multimodal system [88], which instead of supporting a direct human-computer interface for sequences of command/display turns, accumulates ambient perceptual observations during structured multi-party interactions [82] like the creation

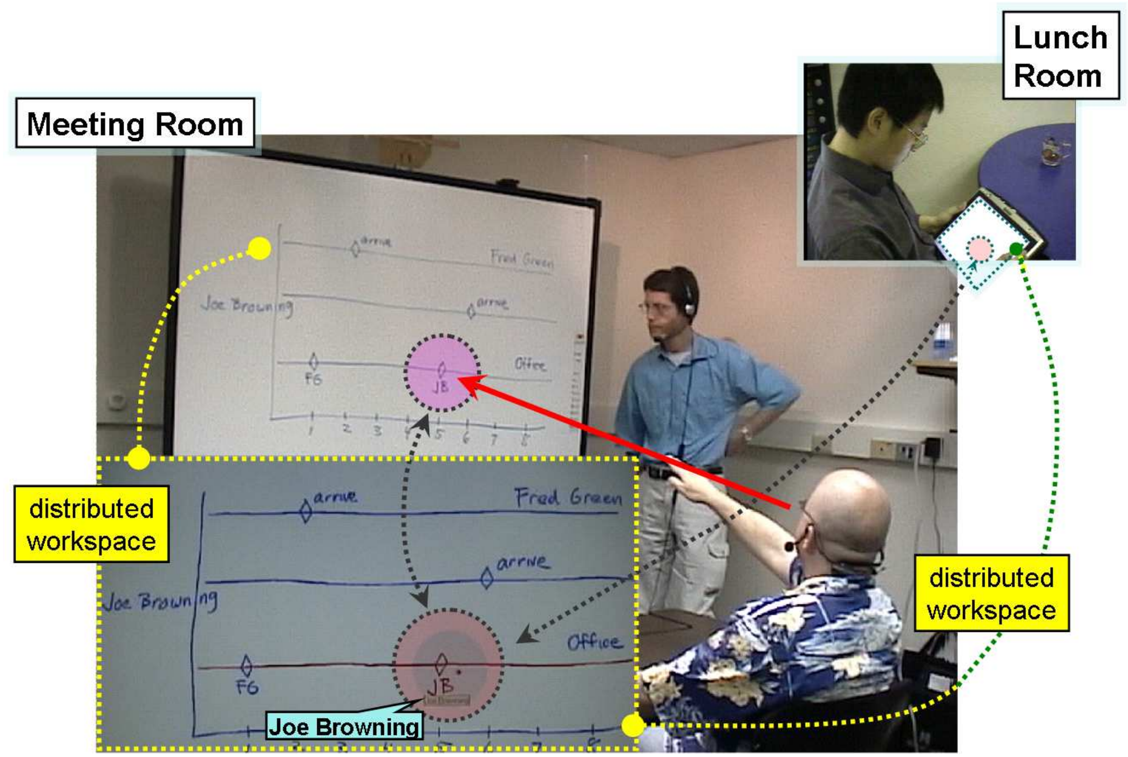


Figure 3.2: Our *ambient-cumulative* Charter system being used to support remote meeting collaboration. SHACER's dynamic learning of both the chart label, *Joe Browning*, and its abbreviation, *JB*, support distributing to the remote user a semantically enhanced representation in the form of a focus bubble (centered on the *JB* milestone) and hover label (explaining the abbreviation). Together these distributed annotations allow the remote user to better understand what the seated user is talking about when he points at the office-space-availability milestone for *Joe Browning* and says, "…there's a problem with *his* office space."

of a Gantt schedule chart during a meeting. The meetings are structured in that a Gantt chart itself has inherent structure, both temporally and spatially. For example drawing the axes temporally precedes drawing tasklines, so speech associated with the creation of the axes precedes speech which is associated with creation of the tasklines. Drawing the chart axes also creates a spatial context that allows the system to judge the difference between tasklines, which are structurally part of the chart, and other spatially extraneous lines like pointers, connectors or underlines.

The ACI we employ for testing SHACER is a suite of agents collectively referred to as Charter. A common characteristic of human-human interactions during which Charter can be deployed is that there are public spaces. For example, there is a shared interactive whiteboard or a piece of digital paper for public sketching and handwriting [90, 16]. There is a shared conversational space for speech captured by close-talking microphones [87, 89]. There is a shared gestural space tracked by stereo-video cameras [94, 51] capable of recognizing deictic pointing or gazing events. The system's function is to unobtrusively collect, recognize, integrate and understand the information it observes in those public spaces, producing useful background artifacts.

Our ambient-cumulative Charter application [88] can, for example, recognize deictic pointing events during a meeting. It can then leverage its background understanding of the meeting's content to distribute a semantically enhanced representation of that pointing event. In the example shown in Figure 3.2, SHACER learned that the *JB* handwriting was an abbreviation for *Joe Browning.* The full name, *Joe Browning*, had been handwritten on the shared distributed workspace a few minute earlier. When it was written it was also spoken redundantly, so SHACER learned and enrolled it. When *JB* was handwritten the writer also said, "Joe Browning." This second redundancy allowed SHACER to make the association between *JB* and *Joe Browning.* Charter distributed that dynamically learned knowledge to aid the remote user's understanding of the meeting. Thus, when the remote participant heard the utterance, "...there's a problem with *his* office space," he also saw a circular focus area over the *JB* milestone. This was triggered by recognition of the deictic pointing gesture. Within the focus area was a hover label holding the expanded meaning of the abbreviation, which had been dynamically learned. In this way the remote participant could better understand whose office was being discussed, and that information was only available due to SHACER's ability to dynamically learn the meaning of a newly introduced abbreviation.

The long range goals of our ambient-cumulative system are quite similar to those of the "ubiquitous computing" and "ambient intelligence" research efforts as described by Pantic [140]. The vision of such research efforts calls for a shift of focus away from computer-centered system design and towards human-centered design [132], which leverages unobtrusive perception to understand context and adapt automatically in anticipation of a user's needs. As shown in Figure 3.2 Charter/SHACER attempts to move in this direction of dynamically adapting to user needs, by learning the meaning of abbreviations and providing them in useful contexts that can aid remote users' understanding. Charter/SHACER can be used to integrate information from three modes of input: speech, handwriting/sketching, and 3D deictic gestures like pointing. In this thesis the examination and analysis of three-way integration of speech, handwriting and 3D gesture is left for future work. The focus here is on the integration of speech and handwriting for learning new vocabulary.

3.2 SHACER PROCESSING

SHACER's goal is acquiring new vocabulary dynamically in context, which is a necessary part of what it means to learn. The first problem in learning is called the *situation-identification* problem: identifying what the relevant features are out of all the possible perceptual cues coming into the system [36]. For example, in language acquisition studies, Baldwin *et al.* [15] have shown that infants more easily link objects with their names when the person speaking the object-name is also looking at or pointing to the object. Thus, multimodal redundancy identifies the relevant perceptual cues and focuses attention on them [61, 12, 175, 177, 150]. This is the theoretical assumption underlying SHACER's approach to learning new words. Multimodal redundancy is the situational identifier that focuses attention on the important features to be learned: the redundantly presented spoken word and its handwritten spelling. Knowing that these are the perceptual cues to be attended to, how can the occurrence of multimodal redundancy be leveraged to help in dynamically learning new words?

3.2.1 Modal Recognition

Figure 3.3 diagrams the input paths of speech and handwriting into the system. This example occurred during the second of the *G* series of meetings. In this *G2* meeting the meeting facilitator drew a Gantt chart to schedule space and equipment for a new hire. He said, "This is our timeline for Fred Green," and also wrote that name on a Gantt chart taskline on the whiteboard. In this example, the spoken two-word name, *Fred Green*, was OOV. Both individual words occurred in the transcribing speech recognizer's vocabulary, but the two-word name as such was neither in the vocabulary nor was it part of the training data on which the recognizer's language model was built. SHACER uses three different types of speech recognizers. The transcribing speech recognizer, referred to here, distinguishes between speech and non-speech acoustic input and produces a transcription of the speech. Because the sequence *Fred Green* was not explicitly accounted for in the language model, the recognizer was forced to choose a sequence that was acoustically very close and that had some probability assigned to it by the language model. This is typically what transcribing speech recognizers do when they encounter OOV terms — force recognition to acoustically similar in-vocabulary words whose sequence has some probability in the language model [69]. In the Figure 3.3 example, the result of this forcing was the insertion of a possessive *s* between *Fred* and *Green*.

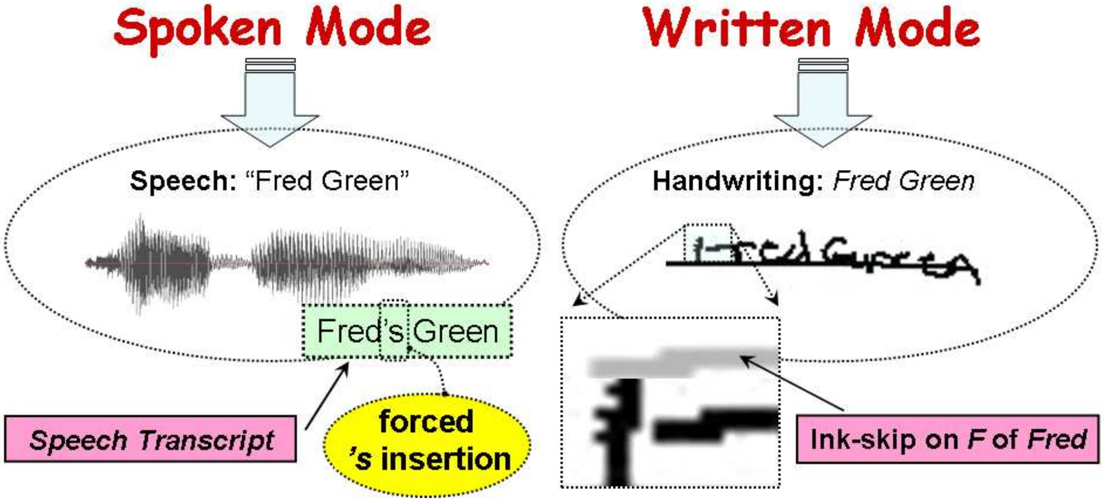


Figure 3.3: Two modes of input to SHACER: (left) spoken input, and (right) handwritten input. The spoken two-word name, *Fred Green*, was OOV, so the transcribing speech recognizer forced recognition to an allowed two word sequence, inserting an *s* between the words *Fred* and *Green* in the process. On the handwriting side, the top-stroke of the *F* in *Fred* was not perceived. It was skipped, making it difficult for the handwriting recognizer to correctly interpret.

Handwriting recognizers also have vocabularies and language models. If letters are carefully handwritten then it is possible that OOV terms can be recognized correctly.

However, words that are in the handwriting recognizer's vocabulary and language model are more easily recognized. In the Figure 3.3 example, handwriting recognition is further confounded by an error in ink perception. The topmost stroke of the *F* in *Fred* was written too lightly on the touch-sensitive whiteboard that was collecting ink during this meeting. So the perception of that stroke was skipped, making it very difficult for the handwriting recognizer to interpret the handwriting correctly.

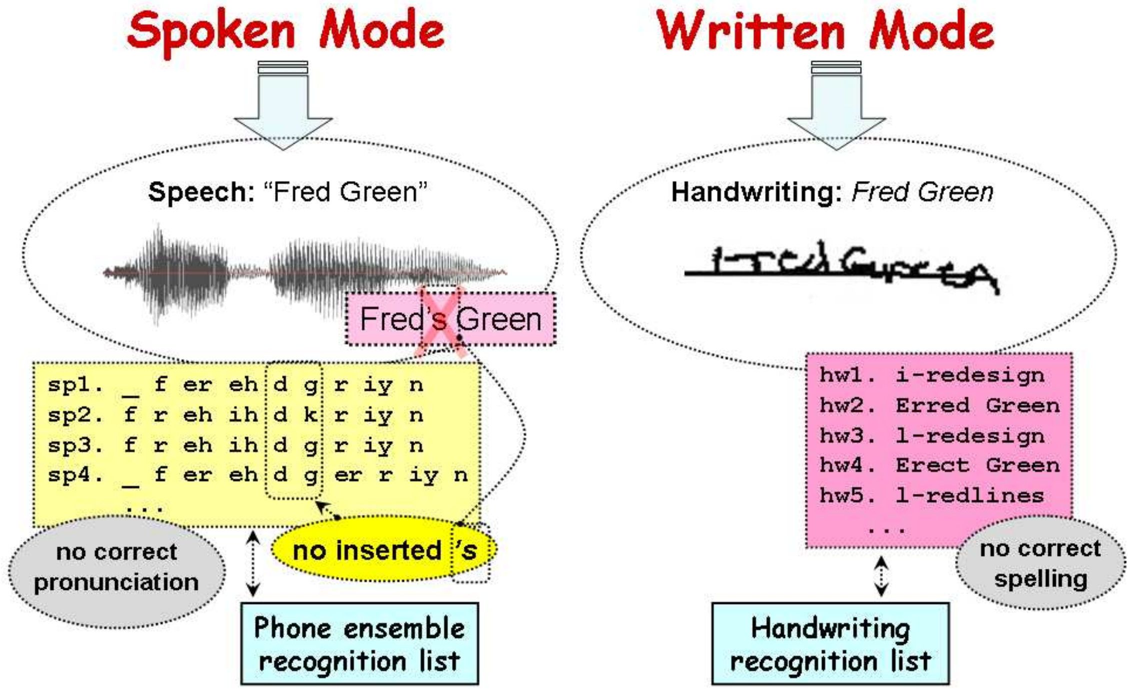


Figure 3.4: On the right is the list of alternate phone-level recognitions of the speech, from an ensemble of four phone recognizers. The correct phone-level pronunciation is not present on the list; however, phone-level recognition does not insert the possessive *s* between *Fred* and *Green* that results from the transcribing recognizer's forced insertion. On the left is the list of alternate handwriting recognizer outputs. Due to the ink-skip the correct spelling does not occur.

The second type of speech recognizer used in SHACER is phone-level recognition. This is illustrated in Figure 3.4, where below the spoken, *Fred Green*, is a list of alternate phone-level recognitions. SHACER uses phone recognizers to mitigate the effect of forced insertions that are imposed by the transcribing recognizer. None of the phone recognizers spuriously insert the "s" between Fred and Green. In this example phone-level recognizers do a better job at acoustically interpreting the OOV proper name, but the canonically correct [2] pronunciation still is not present in the list. This is the primary reason that systems for spoken document retrieval rely on sub-word unit recognizers [123, 125, 31]. For example, Yu *et al.* [181, 180] report that spoken document retrieval based on phone-level processing consistently outperforms word-level based processing, because phonemes provide better coverage of OOV words including OOV query terms.

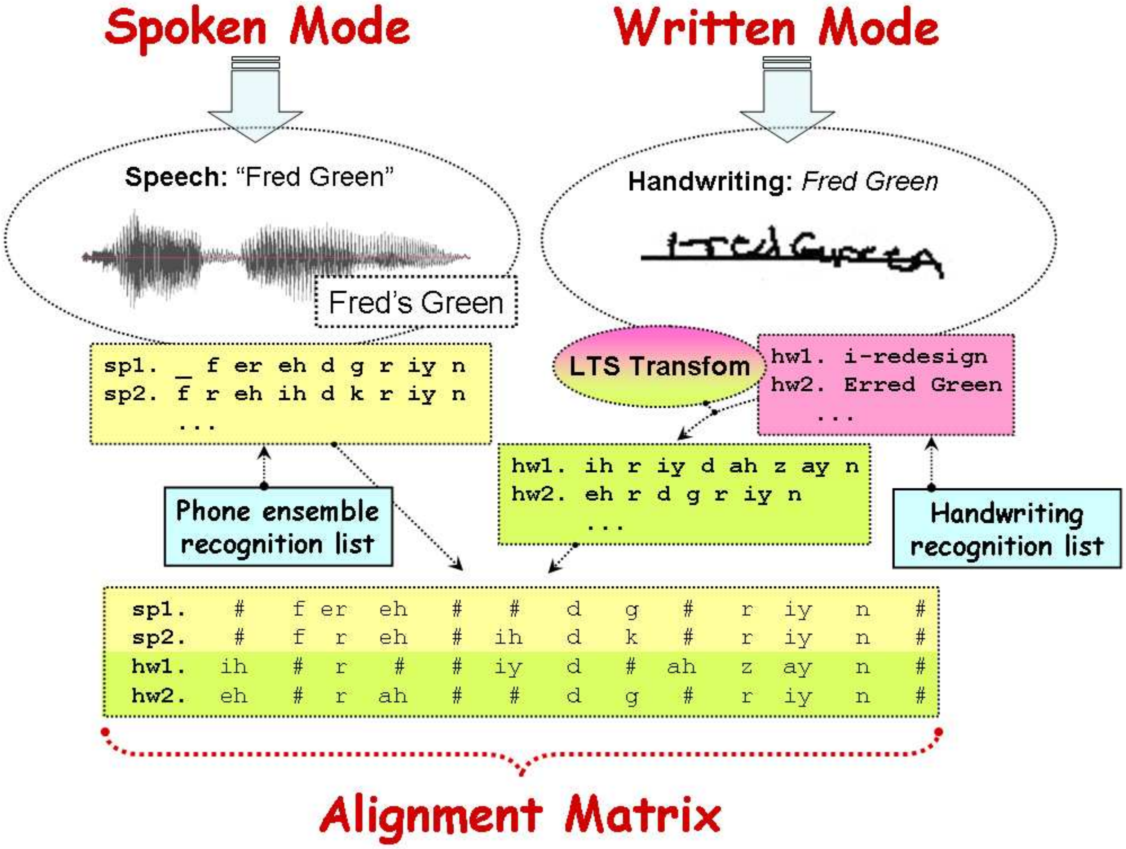
The list of alternate letter sequence interpretations of handwriting input in Figure 3.4 appears below the handwritten *Fred Green*. Because of the ink-skip the correct spelling of the name does not appear in the list.

3.2.2 Phonetic Alignment

The second problem in learning, which is another basic axiom from the field of Artificial Intelligence (AI) [36], is called the *description problem*. In order to know that two situations are similar they must be describable in a similar notation. Without that there is no way to compare them. To detect multimodal redundancy and recognize new words, SHACER's first step is aligning the handwritten words to nearby spoken words. Closely matching alignments then trigger SHACER's detection of multimodal redundancy. To make the handwritten words comparable to the spoken words SHACER transforms the handwriting letter-string alternatives into sequences of phonemes. This process is called Letter-To-Sound (LTS) transformation. It is illustrated in Figure 3.5), and SHACER accomplishes it by using a letter-to-sound transformation module from Alan Black's CMU FLITE toolkit [22]. The resulting phonemes are then aligned against the speech phonemes as shown in the *Alignment Matrix* at the bottom of Figure 3.5.

SHACER's phonetic articulatory-feature based aligner compares phone hypotheses by feature sets rather then by phone name. Instead of assigning the phone match between g and k an absolute score of 0, because they are not the same phone, it can instead assign them a metric that takes into account the fact that they are identical in all articulatory

[2] A canonically correct pronunciation for a word is the phonetic pronunciation listed for that word in a standard dictionary. For this thesis the standard dictionary used is the Carnegie Mellon University (CMU) Dictionary, Version 6.0.



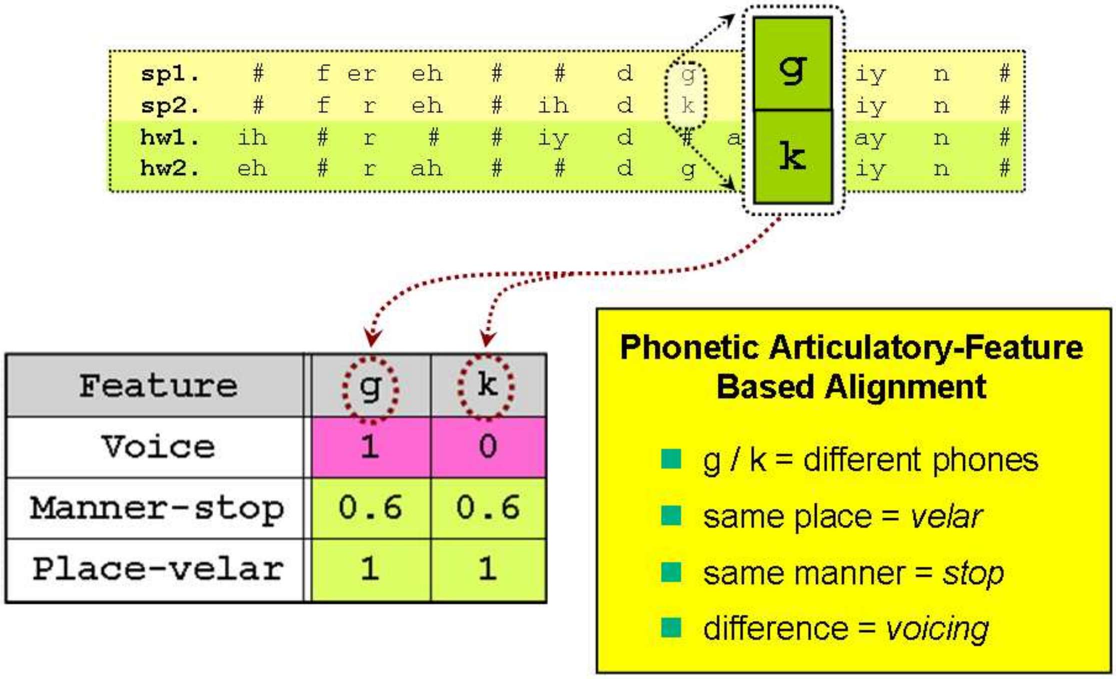
sp1.	#	f	er	eh	#	#	d	g	#	r	iy	n	#
sp2.	#	f	r	eh	#	ih	d	k	#	r	iy	n	#
hw1.	ih	#	r	#	#	iy	d	#	ah	z	ay	n	#
hw2.	eh	#	r	ah	#	#	d	g	#	r	iy	n	#

Figure 3.5: After speech and handwriting streams have been individually recognized, they need to be aligned to check for redundancy. First the handwriting is put through a Letter-To-Sound transform (LTS), which is a transformation of sequences of letters into sequences of phonemes. Then the phone ensemble phone sequences can be aligned with LTS phone sequences.

features except voicing. This is illustrated in Figure 3.6, which highlights the alignment of the g and k phones that are hypothesized by different ensemble phone recognizers to start the word *Green*. Both phones share a similar place and manner of articulation. They are both velar stops.

3.2.3 Pronunciation Refinement

When the alignments of LTS handwritten phone sequences and speech phoneme sequences are close enough, then SHACER treats the respective handwritten and spoken inputs as being possibly redundant. The next step in processing is to use the information embedded



sp1.	#	f	er	eh	#	#	d	g		iy	n	#
sp2.	#	f	r	eh	#	ih	d	k		iy	n	#
hw1.	ih	#	r	#	#	iy	d	#	a	ay	n	#
hw2.	eh	#	r	ah	#	#	d	g		iy	n	#

Feature	g	k
Voice	1	0
Manner-stop	0.6	0.6
Place-velar	1	1

Phonetic Articulatory-Feature Based Alignment

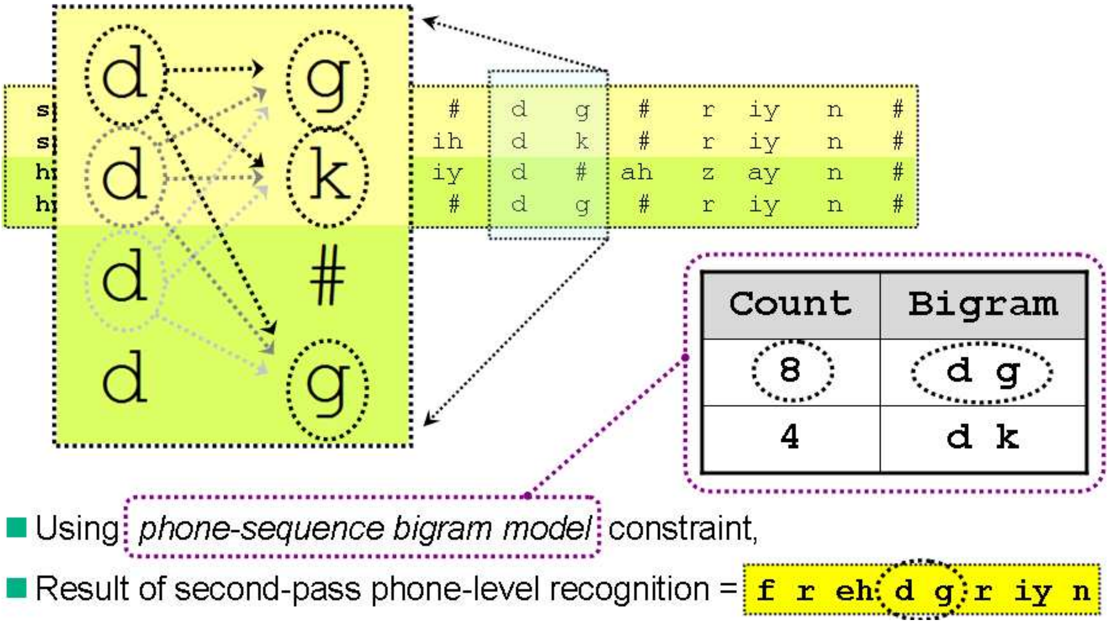
- g / k = different phones
- same place = *velar*
- same manner = *stop*
- difference = *voicing*

Figure 3.6: SHACER uses an articulatory-feature based alignment mechanism, which does not insist that phones must be spelled the same way in order to match. Thus *g* and *k* are aligned here, because they are both velar stops and differ only in that one is voiced while the other is not voiced.

in the alignment matrix to better model the phone sequence transitions that are possible — that is, given one phone what is the most likely next phone based on information in the phonetically aligned columns of the alignment matrix. For example, Figure 3.7 highlights the alignment matrix columns that represent the transition from the final phoneme of the word *Fred* to the first phoneme of the word *Green*. There is some ambiguity as to whether the first phoneme of the word *Green* is *g* or *k*. Counting the phone bigrams across rows at this transition point, as illustrated in the expanded highlight box of Figure 3.7, yields a table of bigram counts. The count of *d*-*g* bigrams is 8, while the count of *d*-*k* bigrams is 4. Based on these bigram statistics it is more likely that the first phoneme of the word *Green* is *g* and not *k*.

For the example shown in Figure 3.7, the resulting bigram sequence model for the entire alignment matrix was used to constrain a second pass phone-level recognition of the speech. In this case that second pass recognition yielded the correct pronunciation.

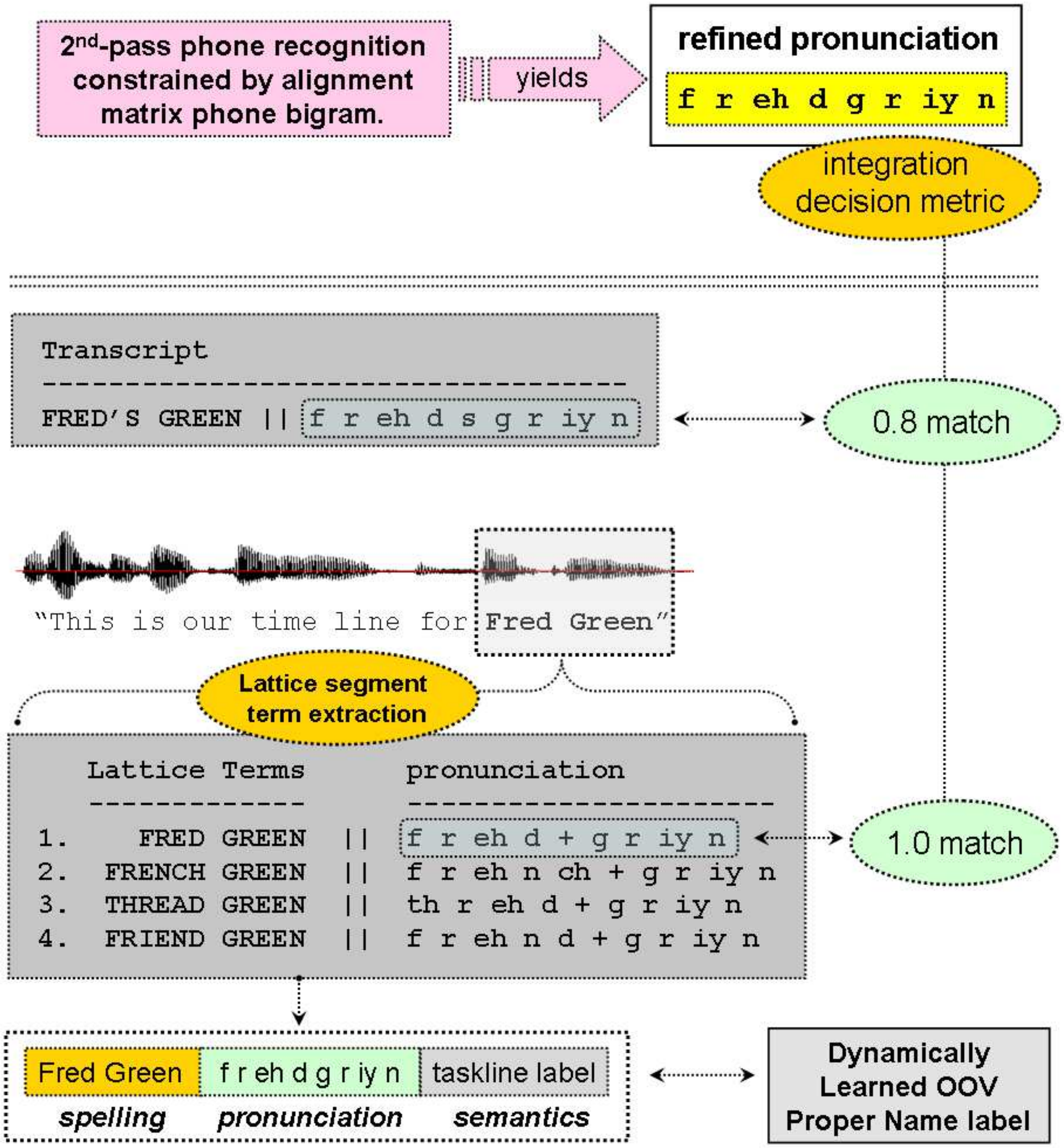
There was no incorrectly inserted possessive *'s* between *Fred* and *Green*, as was the case for the transcribing recognizer. Even though the correct pronunciation of the spoken name, *Fred Green*, appeared neither in the list of ensemble phone alternates nor in the list of LTS phone sequences from the handwriting recognition, nonetheless the articulatory-feature based alignment of the combined inputs correctly discovered the redundancy and provided enough acoustic information to extract the correct pronunciation. This is the benefit of leveraging multimodal redundancy. The redundant modes offer complementary information, which when properly combined can yield better recognition than is possible in either mode alone [85].



Count	Bigram
8	d g
4	d k

- Using phone-sequence bigram model constraint,
- Result of second-pass phone-level recognition = f r eh d g r iy n

Figure 3.7: SHACER uses cross-row phone sequence information from the alignment matrix to create a bigram phone sequence model. This model can combine information from both handwriting and speech phone sequences to resolve ambiguous transitions like that shown here from the last phoneme of *Fred* to the first phoneme of *Green*. Using the model from the entire matrix to constrain a second-pass phone-level recognition yields the correct pronunciation.


2nd-pass phone recognition constrained by alignment matrix phone bigram.
yields
refined pronunciation
f r eh d g r iy n
integration decision metric
Transcript
FRED'S GREEN || f r eh d s g r iy n
0.8 match
"This is our time line for Fred Green"
Lattice segment term extraction
1.0 match
Fred Green
f r eh d g r iy n
taskline label
spelling
pronunciation
semantics
Dynamically Learned OOV Proper Name label

	Lattice Terms		pronunciation
1.	FRED GREEN	||	f r eh d + g r iy n
2.	FRENCH GREEN	||	f r eh n ch + g r iy n
3.	THREAD GREEN	||	th r eh d + g r iy n
4.	FRIEND GREEN	||	f r eh n d + g r iy n

Figure 3.8: The refined pronunciation produced by constrained second-pass phone recognition is used as an *integration decision metric* against which to measure interpretations from all input sources. The closest sources are chosen to represent the spelling and pronunciation of the new word. Comparisons are shown for the speech transcript (non-matching), versus word sequences extracted from the temporally corresponding segment of the speech recognizer's lattice — which in this case result in an exact pronunciation match.

3.2.4 Integration

The refined pronunciation resulting from constrained second-pass phone recognition then becomes the key to how SHACER leverages multimodal redundancy. That refined pronunciation is used as a metric against which to measure hypotheses from all input sources, as shown in Figure 3.8. When it is compared against the transcript, it does not match exactly. It is also possible to use temporal boundary information from the alignment matrix to choose segment of the transcribing speech recognizer's lattice from which local word sequences can be extracted. In this example, when the refined pronunciation is compared against those extracted word sequences the acoustically most likely word sequence was found to be an exact match.

This exact match is very strong evidence that, *Fred Green*, was in fact what was spoken and written; so, that new spelling, pronunciation and semantics are dynamically enrolled in the system.

3.2.5 Learning

The are two results of learning in SHACER (Figure 3.9). The first result is immediate — the incorrectly recognized Gantt chart label (i.e., *i-redesign*) for *Fred Green* is corrected. The fact that *Fred Green* is meant to be a taskline label, which in this context is the learned *semantics* of the new term, also participates is this immediate aspect of learning. For example, because of the underlying ink's temporal and spatial relation to the Gantt chart axes and taskline it is treated as a taskline label and not as a milestone label or other extraneous handwriting. The second result of learning in SHACER is persistent system enrollment. The primary enrollment site is the dictionary and language model of a dedicated word/phrase-spotting speech recognizer. Later, when the enrolled word is uttered again, the word/phrase-spotter will recognize it with high confidence. Thus enrollment persists both within a meeting and across meeting boundaries, while immediately learned labels improve recognition but do not of themselves (i.e., without enrollment) have a persistent effect on later recognition. Currently SHACER avoids falsely recognizing new

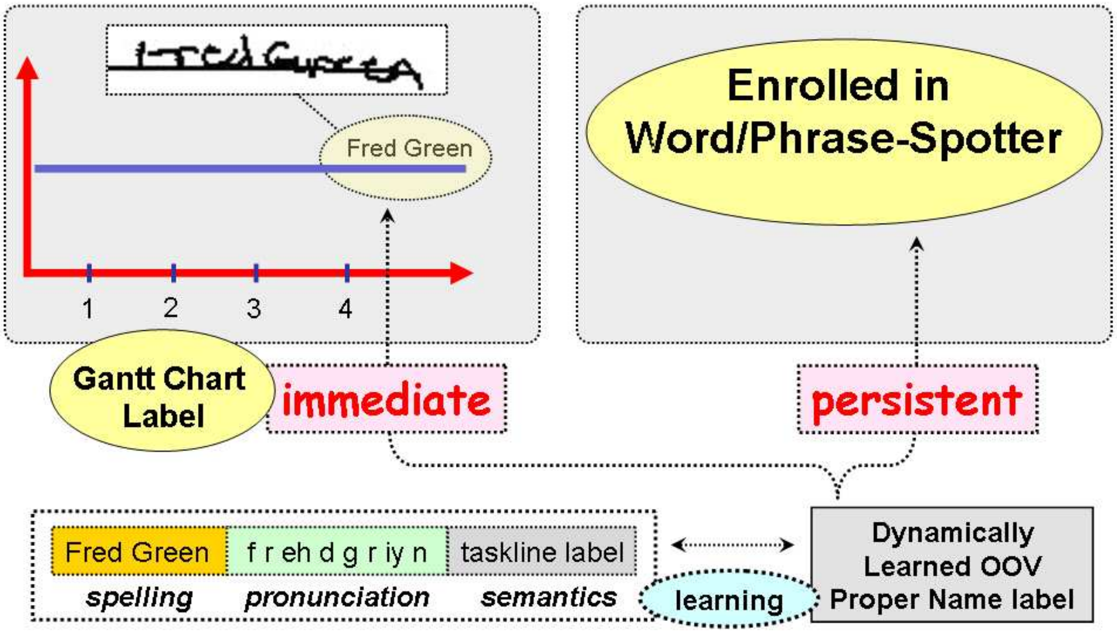


Figure 3.9: There are two aspects of learning in SHACER. *Immediate* learning serves to correct mis-recognitions, as is the case for the *Fred Green* Gantt chart label. *Persistent* learning is the result of enrollment — the spelling and pronunciation of new terms are added to the dictionary of a dedicated Word/Phrase-Spotting speech recognizer. Subsequent recognition of the new term is thus improved.

word spellings and pronunciations by using a high threshold on the likelihood of a detection and integration. If incorrect spellings or pronunciations are enrolled then currently they persist until replaced by a more likely spelling and pronunciation combination.

3.2.6 Understanding Abbreviations

Multimodal redundancy can also help in learning abbreviations. Figure 3.10 shows some inputs that occurred during the fourth of the *G* series of meetings. The discussion above about the persistent enrollment of the newly learned name, *Fred Green*, happened in meeting *G2* — the second of the *G* series of meetings. Along with *Fred Green* another new name was also learned in the *G2* meeting, *Cindy Black*.

Ink and speech inputs, along with persistent learning, are entering the system at the bottom of this Figure 3.10's diagram. The perceived inputs are (1) the two spoken utterances, "Fred Green," and, "and Cindy Black," and (2) the sketch/handwriting ink, which in this case is a diamond shaped Gantt chart milestone symbol written on a taskline and two hand-lettered abbreviations (i.e., *FG*, *CB*) listed below it.

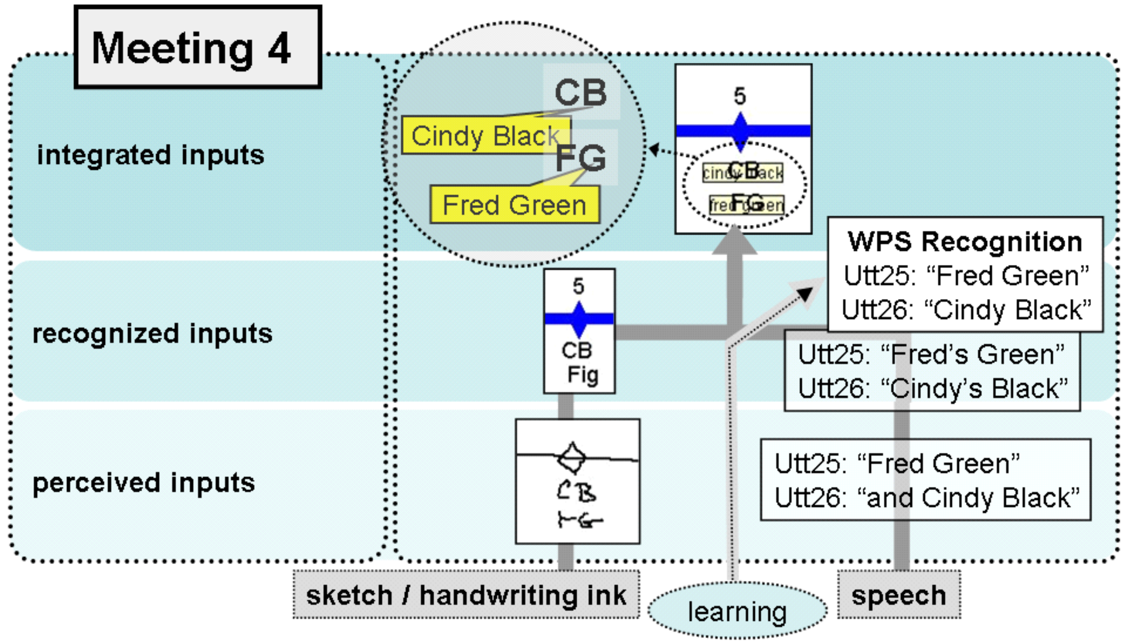


Figure 3.10: Understanding abbreviations in SHACER: a facilitator writes a diamond shaped milestone on the whiteboard Gantt Chart, writes two abbreviations below it (i.e., *CB*, *FG*), and also says, "Cindy Black and Fred Green are arriving here at week five." Given these inputs and the previous enrollment of *Cindy Black* and *Fred Green* the system corrects and expands its understanding of the abbreviations.

These perceived inputs are recognized, as shown in Figure 3.10's *recognized inputs* panel. The handwriting recognition gets the letter sequence for *CB* correct, but the letter sequence for *FG* wrong. At this point, the system has no idea what relationship these letter sequences may have to the two spoken utterances. Thus the abbreviation letter sequences are ungrounded.

Both proper name utterances depicted in Figure 3.10's *recognized inputs* panel are unrecognizable sequences for the transcribing speech recognizer, because neither are listed as two-word names in either the recognizer's dictionary or its language model. Thus the resulting speech recognition transcripts, *Fred's Green*, and *Cindy's Black*, are both incorrect. However, these names were both learned and enrolled earlier in the *G2* meeting, as discussed above for *Fred Green*, so both OOV names are correctly recognized by the

Word/Phrase-Spotter — as shown in the *WPS Recognition* box in Figure 3.10's *recognized inputs* panel.

The Word/Phrase-Spotter recognitions of *Cindy Black* and *Fred Green* trigger a search for temporally nearby handwriting, and their associated spellings are compared to any handwriting that is found. For the cases shown in Figure 3.10's *integrated inputs* panel, the handwriting instances (i.e., *CB* and *FG*) are first-letter abbreviations of the spoken, newly enrolled names. SHACER currently can recognize first-letter and prefix abbreviations. In the future other forms of abbreviations will be recognized also. These abbreviations can then be associated with their expanded semantics: *FG* = *Fred Green*, and *CB* = *Cindy Black* [87], as shown by the hover labels in Figure 3.10's *integrated inputs* panel.

SHACER's process of learning abbreviations is called *Multimodal Semantic Acquisition.* The learned semantics carried in one mode, like the WPS speech recognitions of *Fred Green* or *Cindy Black*, are dynamically acquired by new symbols in another mode, which in this case are the handwritten abbreviations, *CB* and *FG*. Thus unknown handwritten abbreviations, which are redundantly spoken, are grounded by acquiring their expanded meanings from previously recognized and enrolled speech.

3.3 SUMMARY

We have shown that SHACER can learn new words during meetings by learning them dynamically. The chapter began by pointing out that in natural speech contexts both proper names and handwritten abbreviations occur frequently, are critical for understanding, and are also more likely than other words to be out-of-vocabulary. This raises the need to learn OOV terms dynamically. This chapter has given a high-level overview of how SHACER accomplishes dynamic learning of OOV terms.

SHACERs method depends on leveraging the occurrence of multimodal redundancy. In that context, we have outlined SHACER's three main functionalities.

1. **Alignment:** SHACER uses an articulatory-feature based alignment mechanism for detecting redundancy. Phonetically close alignments of speech and letter-to-sound transformed handwriting are processed as possible redundancies.

2. **Refinement:** Using a phone-bigram sequence model derived from the alignment matrix, SHACER produces a refined pronunciation hypothesis for a new term. Even when neither individual input mode yields the correct pronunciation alternative, combining their information by using the alignment matrix to constrain second-pass recognition allows the system to still recover the correct pronunciation.

3. **Integration:** Using the refined pronunciation as an *integration decision metric* against which to compare other inputs, SHACER can decide on the best combination of spelling and pronunciation. Integrating information from speech and handwriting can yield better recognition and understanding than is possible in either mode alone.

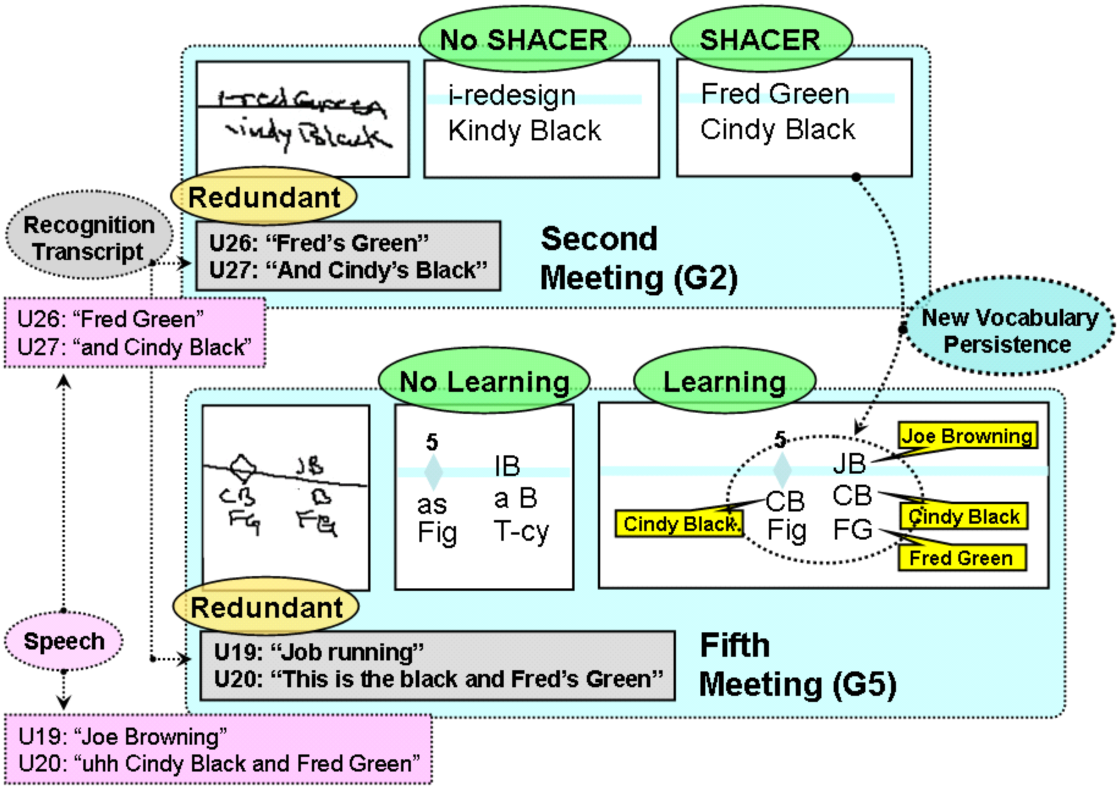


Figure 3.11: An overview of the SHACER's learning and understanding capabilities.

Figure 3.11 summarizes the capabilities described in this chapter. In the *G2* meeting, when two new proper-name Gantt chart labels were introduced redundantly, the recognition in both modes failed. Thus without SHACER both labels were incorrect. With

SHACER, leveraging multimodal redundancy, both labels were correctly recovered and enrolled. In the *G5* meeting, there were many abbreviations. With no learning none of the five abbreviations shown in Figure 3.11 could be recognized or understood. With learning four of five of the abbreviations were recognized and understood correctly. They were both correctly spelled and correctly understood in terms of their expanded meanings (as shown by the hover labels in Figure 3.11). This was possible because all of these abbreviations were spoken redundantly, and all referred to terms that had been dynamically learned in an earlier meeting. As enrolled terms they persisted across meeting boundaries in the dictionary and language model of the dedicated Word/Phrase-Spotting recognizer, and were recognized when spoken again.

In the following chapters we will discuss the three main steps of SHACER processing — alignment, refinement and integration — in more depth and give detailed examples. Finally we will present test results on a held-out data set, which make it evident that SHACER can yield significant improvements in understanding.

Chapter 4

Prelude to SHACER: Multimodal New Vocabulary Recognition

4.1 HIGH LEVEL GOALS

4.1.1 Dynamic Learning

Machines are moving closer to being observant and intelligent assistants for humans [25, 9, 23, 55]. However, multimodal and spoken dialogue systems are typically implemented with fixed vocabularies and knowledge spaces. Their automatically acquiring new knowledge as they are running, particularly by a single, natural demonstration would significantly enhance the usability of such systems. The dynamic augmentation of vocabularies, pronunciation lexicons and language models is an active area of research in speech and gesture recognition [39, 40, 41, 147, 92, 144]. Machines or systems that assist humans in real-time [1] tasks need to be able to learn from being shown — through sketch [38, 154], handwriting [100], teleassistance [143], speech [162], or multimodally through handwriting and speech as in the work described in this chapter.

One example of learning from being shown is understanding abbreviations. Often the handwritten terms during a whiteboard presentation are abbreviated, like those in Figure 4.1. During the presentation their meaning is grounded in speech, and thus clear to listeners. Listeners can learn an abbreviation's grounding quickly and easily as they see it written and hear it spoken. Such redundantly grounded terms then seed subsequent

[1] Real-time: a descriptive phrase that refers to events simulated by a computer at the same speed that they would occur in real life.

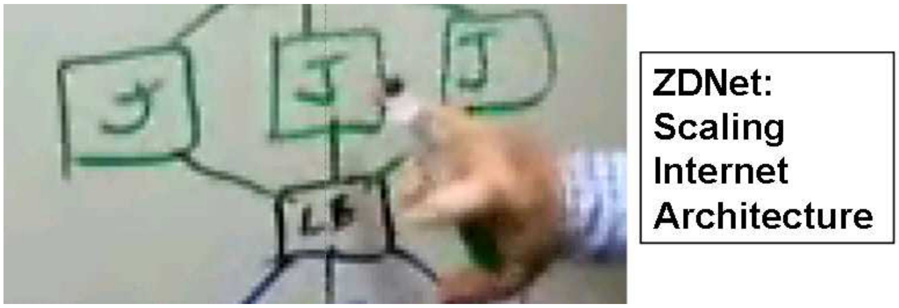


Figure 4.1: These abbreviations, *J* and *LB*, are critical to understanding this presentation. During the presentation they are spoken was well as being written. The speech grounds their meaning. Out of context it is difficult to know their meaning. A useful function of an assistive computational program would be to perceive and recognize that meaning during the presentation, so it would be available later.

semantic entrainment [43, 27]; that is, they are used over and over again with the same understood meaning. They become the dialogue vernacular, the common talking points. They are repeated throughout the presentation. For example, the abbreviation *J* in Figure 4.1 was redundantly entrained to represent the term *Java tier*. In a presentation about *Scaling Internet Architecture* that is a critical term. The abbreviation *J* was written with that same meaning five times during the four minute presentation and referred to eight times. However, without having watched the presentation, it is difficult to just look at the diagram and know what the *J* or *LB* abbreviation below it refer to. If an assistant program could leverage the fact that when the abbreviation was written on the whiteboard it was also spoken, then it could learn that association and annotate the diagram with explanatory hover labels — as shown in Figure 4.2. These hover labels identify the expanded contextual meaning of the abbreviation.

In later chapters we will show how SHACER, building on the system, ideas and outcomes discussed in this chapter, can perceive and recognize abbreviations that are redundantly spoken as they are handwritten, and thus provide the dynamically learned semantic labels shown in Figure 4.2 (*J* = *Java tier*, *LB* = *Load Balancer*). The work in this chapter is a prelude to SHACER. It leverages multimodal redundancy across handwriting and speech to dynamically learn out-of-vocabulary words. The system is called Multimodal

New Vocabulary Recognition (MNVR). This chapter lays the foundation for our later describing what SHACER does. It shows that combining information from redundant handwriting and speech leads to significantly better recognition than can be achieved in either mode alone. MNVR is about leveraging multimodal redundancy for better recognition, and that functionality is what it shares with SHACER. Leveraging multimodal redundancy is the functionality underlying the dynamic abbreviation understanding shown in Figure 4.2.

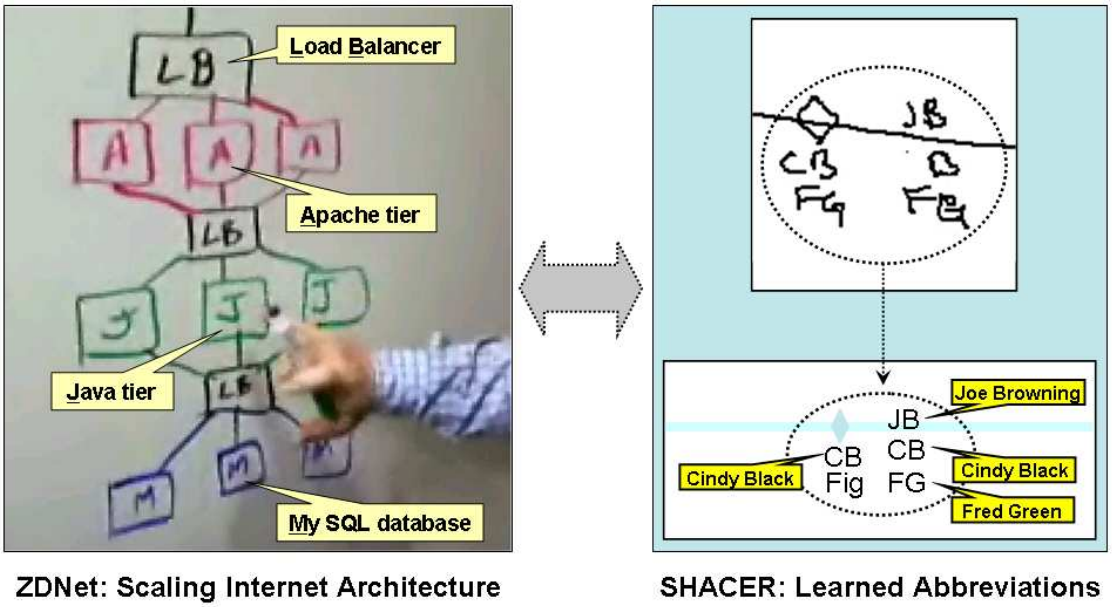


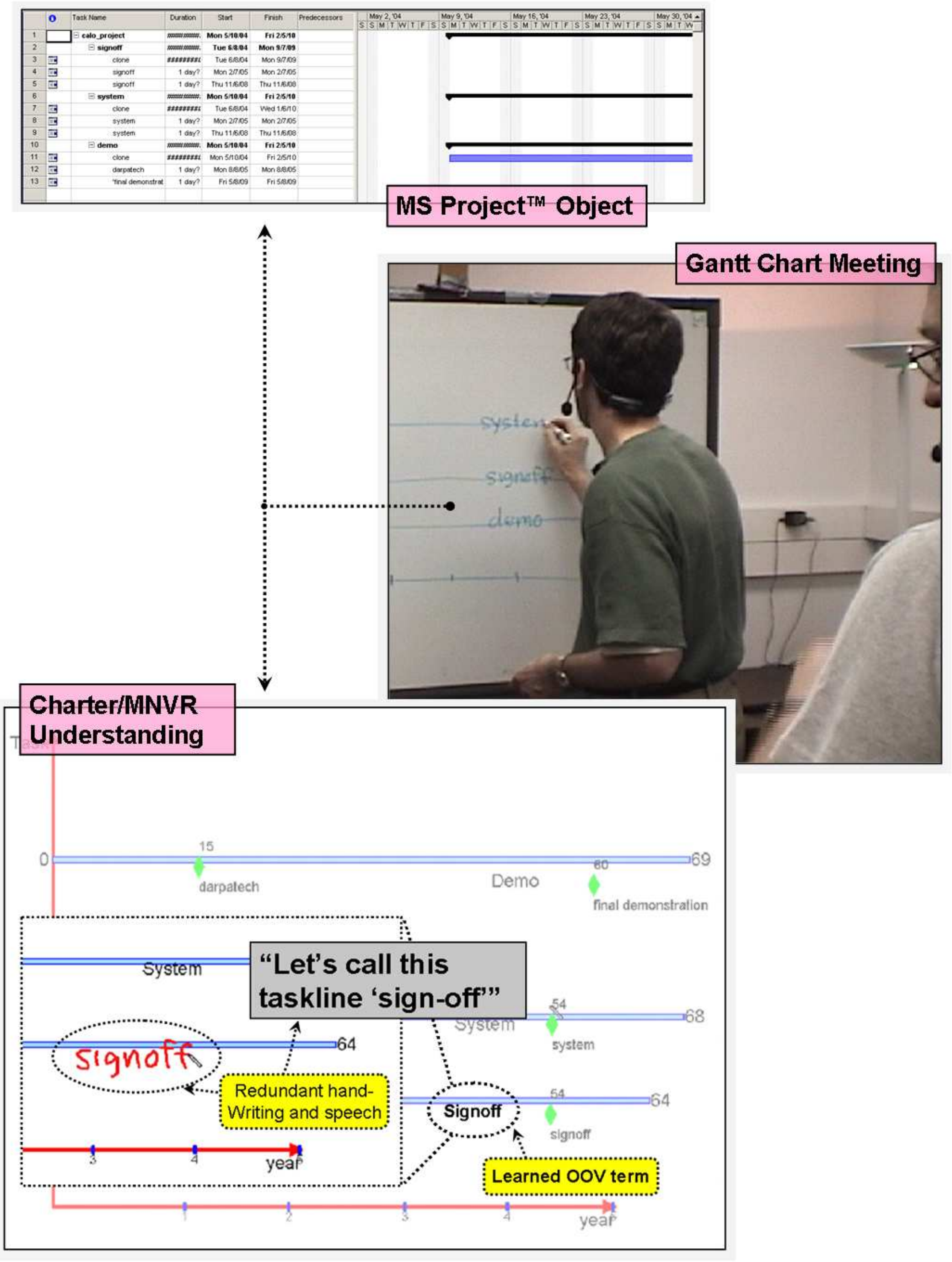
Figure 4.2: Instances of multimodal redundancy serve as vernacular anchor points; that is, they focus attention when they are introduced so that subsequent presentation structures can be understood — like the flow diagram on the left, taken from a ZDNet *At the Whiteboard* presentation on *Scaling the Internet Architecture*. SHACER has been designed to address the need for this capability. Eventually the dynamically learned, informative hover labels displayed in SHACER's processed Gantt chart (right side) could also be available in applications like the ZDNet presentation on the left.

Our aim, as for Breazeal *et al.* in their work on designing humanoid robots to be cooperative partners for people, is that our system will be able to "acquire new capabilities . . . as easy and fast as teaching a person" [25]. To take a first step in this direction our MNVR technique focused on a single, important capability within the scope of what humans ultimately need to teach a cooperative machine: establishing a common, working vocabulary of spoken words taught to the machine by natural demonstration as the system is running.

4.1.2 The Vocabulary Problem

Most computer systems require users to type or speak the right words. For example the newly released Microsoft Vista ^{TM} Operating System completely speech-enables the windows command interface. Any menu item or any button name can be spoken — but must be spoken verbatim to be recognized. Dozens of text manipulation commands in speech dictation mode must also be memorized verbatim in order to be used. For example, during spoken dictation, one can say, "select *verbatim*," to highlight the word *verbatim* in the previous sentence; but, one cannot say, "highlight *verbatim*," or even, "choose *verbatim*." The problem with this is that users — particularly new or intermittent users — often use the wrong words. This is an aspect of the classic vocabulary problem [56]. In studies of information retrieval searches, users seldom used the same word to refer to a particular concept. Even a set of the 15 most common aliases for a concept covered only 60-80% of the search vocabulary people chose for that concept. Users believe the words they choose are the right ones. Those who are new to an interface or use it only intermittently can grow discouraged or frustrated when the system does not understand their words. For some users, this hurdle is enough to block acceptance of a new interface. Having interfaces that could adapt dynamically to users' chosen vocabulary could help in addressing this issue by taking the burden of learning off the user and shifting it to the system.

The MNVR approach combines handwriting recognition and out-of-vocabulary speech recognition, to leverage two of the richest communicative modes we as humans have available for acquiring new vocabulary. Others have designed OOV speech recognition systems [57, 39, 20, 8, 111], but they are not used in a multimodal context. Related multimodal systems that extract words from statistical associations of object/phone-sequences or action/phone-sequences [176, 62, 150] do not leverage the grammatical and linguistic context in the same way MNVR does, nor do they use handwriting as an input. The key components of MNVR's approach are (1) highly constrained, real-time out-of-vocabulary



		Task Name	Duration	Start	Finish	Predecessors
1		⊟ calo_project	######.######.	Mon 5/10/04	Fri 2/5/10	
2		⊟ signoff	######.######.	Tue 6/8/04	Mon 9/7/09	
3		clone	########	Tue 6/8/04	Mon 9/7/09	
4		signoff	1 day?	Mon 2/7/05	Mon 2/7/05	
5		signoff	1 day?	Thu 11/6/08	Thu 11/6/08	
6		⊟ system	######.######.	Mon 5/10/04	Fri 2/5/10	
7		clone	########	Tue 6/8/04	Wed 1/6/10	
8		system	1 day?	Mon 2/7/05	Mon 2/7/05	
9		system	1 day?	Thu 11/6/08	Thu 11/6/08	
10		⊟ demo	######.######.	Mon 5/10/04	Fri 2/5/10	
11		clone	########	Mon 5/10/04	Fri 2/5/10	
12		darpatech	1 day?	Mon 8/8/05	Mon 8/8/05	
13		'final demonstrat	1 day?	Fri 5/8/09	Fri 5/8/09	

Figure 4.3: A two-person Gantt chart scheduling meeting. The person at the whiteboard is labeling tasklines. Charter's processed understanding of the Gantt chart is shown in the lower left, with an illustration of how MNVR dynamically learns OOV labels. The associated Microsoft Project ^{TM} chart is shown at the top.

speech recognition, (2) standard handwriting recognition , and (3) a multimodal task domain capable of assigning semantics on the basis of spatial, temporal and in some cases linguistic aspects of the input signals (depicted in Figure 4.3). In our task domain the system functions as a real-time, multimodal interface to Microsoft Project $^{\text{TM}}$ [82]. Recognition of multiple input modes (e.g. speech, 2D pen, handwriting, etc.) allows the system to dynamically build a Microsoft Project $^{\text{TM}}$ chart as the meeting proceeds. OOV constituent names, like the task-line labels show in Figure 4.3, are recognized in real-time and enrolled as part of the Microsoft Project $^{\text{TM}}$ chart [82].

4.2 RELATED WORK

A well known approach to leveraging multimodal redundancy is audio/visual speech recognition. Systems that augment speech recognition by visually extracted face and lip movement features [121] employ an early-fusion approach that combines both input streams in a single feature space. Previous work in our group [84, 91] as well as our MNVR technique instead employs a late-fusion approach that combines the output of separate modes after recognition has occurred. For our test-bed schedule-chart application early-fusion is problematic because the temporal relation between handwriting and the speech segments associated with it is neither completely simultaneous nor completely consistent (see Section 2.2.3). It is not completely simultaneous because handwriting is not exactly synchronous with redundant speech (see Section 2.2.3). Some phones may be spoken synchronously as they are written, but that is not usually the case. Handwriting and redundant speech only start simultaneously in 3% of cases, and in almost no cases were they found to both start and stop at the same time. Most of the time handwriting precedes and overlaps speech. Sometimes speech precedes handwriting. They overlap each other in 73% of cases. They are sequential in 24% of cases, meaning that the start of one follows the end of the other. Thus, there is no clear way to know which handwriting goes with which speech until they have been recognized and aligned, and that makes early fusion problematic. Handwriting and speech lack the aligned temporal boundaries that exist between articulated phonemes and visemes, which rely on the same muscle movements.

Only in situations where the system can assume that users have spoken exactly what they have handwritten is there a possibility of early fusion. In that situation no alignment is necessary. The features of the first-occurring stream could be used directly in recognition of the second stream. However, this would require that the handwriting be spoken verbatim, and statistics from Section 2.2.3 show that of the 97% of handwriting events that are spoken redundantly, in only about three-quarters of those cases are the handwritten words spoken exactly verbatim.

4.2.1 Hybrid Fusion for Speech to Phone Recognition

A third possibility, used by MNVR, is a hybrid re-recognition approach that takes initial recognition results from all input modes, and then uses information from one input mode to constrain a subsequent re-recognition pass on the input from another mode. A variation of this approach has been used by Chung *et al.* [39] in their speak and spell technique that allows new users to enroll their names in a spoken dialogue system. User's first speak their name and then spell it, in a single utterance. Thus, there is a single input mode (speech) but separate recognition passes: the first pass employs a letter recognizer with an unknown letter-sequence model, followed by a second pass OOV recognizer constrained by a sub-word-unit language model and the phonemic mappings of the hypothesized letter sequences from the first pass. On a test set of 219 new name utterances this system achieves a letter-error-rate (LER) of 12.4%, a word-error-rate (WER) of 46.1%, and a pronunciation-error-rate (PER) of 25.5%.

To recognize OOV terms MNVR uses simple phoneme sub-word units. Other speech recognition systems use more sophisticated sub-word units. The sub-word-units used by Chung et al for modeling OOV words are those of Bazzi [20]. These are multi-phone sub-word units extracted from a large corpus with clustering techniques based on a mutual information (MI) metric. Bazzi [19] shows that using MI generated sub-word-units outperforms a system that uses only syllabic sub-word units; however, it is interesting to note that 64% of his MI sub-word units are still actual syllables. Chung *et al.* extend the space of sub-word units by associating sub-word-unit pronunciations with their accompanying spellings, thereby making a finer grained, grapho-phonemic model of the sub-word-unit space.

Galescu [57] uses an approach similar to Chung *et al's* in that he chooses grapheme-to-phoneme correspondences (GPCs) as his sub-word-units. He uses an MI mechanism like Bazzi's to cluster multi-GPC units (MGUs). His language model (in which MGUs are treated as words) was trained on 135 million words from the HUB4 broadcast news transcriptions, with MGUs first being extracted from the 207,000 unique OOV occurrences in that training data. He tested OOV word modeling on the individual OOV terms occurring in 186 test utterances, yielding between a 22.9% — 29.6% correct transcription rate, and between a 31.2% — 43.2% correct pronunciation rate. Applying the OOV language model to the complete utterances in the 186 instance test sets yielded a false alarm rate of under 1%, a relative reduction in overall WER of between 0.7% — 1.9%, with an OOV detection rate of between 15.4% — 16.8%. For a large vocabulary system these are encouraging results: there is a reduction in WER, whereas other systems report increases in WER.

In designing the algorithm for OOV recognition and multimodal new vocabulary enrollment within MNVR we chose not to use GPCs because they require a large training corpus, whereas MNVR's static syllable grammar requires none. Since there is evidence that many if not most MI extracted clusters are actual syllables (64% in Bazzi's work), we felt that the loss in recognition accuracy was balanced out by the savings in not having to acquire and process a task-specific corpus.

4.2.2 Multimodal Semantic Grounding

Roy [148] developed robotic and perceptual systems that can perceive visual scenes, parse utterances spoken to describe the scenes into sequences of phonemes, and then over time and repeated exposure to such combinations extract phonetic representations of words associated with objects in the scene. We refer to this process as multimodal semantic grounding.

In these experiments [148, 147, 150, 149] Roy grounds language in sensory-motor experience. Roy describes lexically defined semantics as "ungrounded speech understanding."

The lexicon words acquire meaning by assignment, instead of being grounded in perception. This is the common approach to automatic speech recognition/understanding currently. The intent of Roy's work is to move from lexically defined semantics to perceptually grounded [68] semantics. In order to take some first steps in that direction Roy offers an analysis of work done in the field of early language acquisition by children, pointing out that (1) the learning problem can be solved "without labeled data since the function of labels may be replaced by contextual cues in the learner's environment," and that (2) non-linguistic context can be used to disambiguate words and utterances. The process of using non-linguistic cues is at the crux of early language acquisition by humans. Roy argues that it may well prove to be just as critical for language learning by machines. Roy uses computer analyzed video of parent/child interactions to spot non-linguistic cues like the presence of certain objects. MNVR takes advantage of very specific and informative non-linguistic information in the form of handwriting.

As characterized by Yu and Ballard [176], Roy's work [150] posits a theory of *associationism* at the core of language learning. Roy's model of Cross-channel Early Lexical Learning is called CELL. In experiments with CELL, Roy presented pictures of childrens' play objects (e.g., key, shoe, truck, dog, etc.) along with spoken utterances to CELL's robotic processor. The utterances were automatically extracted from recordings of caregiver/child interactions involving the pictured objects. A lexicon of object-picture/spoken-utterance pairs was extracted using an information theoretic measure. The learning problem was challenging because caregiver speech contained overt references to the play objects only 30% of the time. For example, caregivers often used phrases such as, "Look at it go!" while playing with various objects like a car or ball, etc. Even given this challenging input CELL successfully demonstrated perceptually grounded word learning.

Rather than using string comparison techniques for measuring the similarity between two speech segments (represented as phone-sequences), Roy generated an HMM based on a segment's best phone-sequence representation. Then each segment's speech was passed through the other segment's HMM. The normalized outputs were then combined to produce a distance metric. Of the words extracted by this method with audio only input only 7% were lexically correct, while with both visual and audio input (combined through a further Mutual Information measure) 28% of the words extracted were lexically correct, and of those, half were correct in their semantic association with the visual object. In related work Gorniak and Roy [62] use these techniques to augment a drawing application with an adaptive speech interface, which learns to associate segmented utterance HMMs with button click commands. Instead of unambiguous button-click commands, MNVR associates OOV speech with ambiguous handwriting recognition of Gantt chart constituent labels.

Yu and Ballard [176] point out that *associationism* as the basic mechanism of perceptually grounded semantics is unlikely to be the whole story (see also [61]). Not only do experiments show that speakers have a strong tendency to look at the objects they're referring to [48]; but also, social cognitive skills, such as the ability to understand referential intent, have also been shown to be a major constraining factor in language acquisition [15]. The literature supports the view that even pre-verbal children have already acquired an ability to associate deictic body gestures with intended referents; moreover, both adults and children appear to share this same mechanism. To buttress their claims for the role of *embodied intention* in language acquisition they perform an experiment in adult second-language word and semantic acquisition. Monolingual English participants are shown video of a children's story about animals being read in Mandarin Chinese. Some are shown only the book pages corresponding with the read audio, while others see the reader's gaze tracked by a superimposed dot, based on eye-tracking measures, that indicates what animal on the page the reader is attending to while speaking. The second group who see the intention-cue (the eye-gaze dot) perform significantly better at both word segmentation and semantic association tasks.

The experiment is then repeated with the users replaced by a computer program. The program performs phoneme recognition on the read input, and then uses a temporal proximity measure between possible object references (e.g., animals appearing on the current page) and phone-sequences from the read speech to place phone strings into bins. The binned strings are then processed to extract word-like units that are clustered using an agglomerative clustering routine. The cluster centroids are then associated with referents.

Finally an Expectation Maximization algorithm is used to discover the most likely word-object pairs. The computer program also receives input with and without intentional cues, and again it is found that the presence of intentional cues significantly improves performance.

To further explore *embodied intention* Yu and Ballard [176] developed an intelligent perceptual system that can recognize attentional focus. It uses velocity and acceleration-based features extracted from head-direction and eye-gaze sensor measurements, together with some knowledge of objects in the visual scene. Visual scene analysis is based on head-mounted scene cameras. Within that context, measurements of the position and orientation of hand movements (tracked by tethered magnetic sensor) are used to segment spoken utterances describing the actions into phone-sequences associated with an action. The types of actions were stapling papers, folding papers, etc. Over time and repeated associations phonetic representations of words describing both the objects and the actions performed on those objects was statistically extracted.

Rather than using individual HMMs as the basis of measuring distance between phonetic sequences (as Roy does), Yu and Ballard use a modified Levenshtein distance [103] measure based on distinctive phonetic features. Their use of articulatory features as a basis for phonetic distance measures was inspired by the work of Kondrak [95, 96]. In 960 utterances (average six words per utterance) Yu and Ballard identify 12% of the words as either action verbs or object names that their system attempts to pair with meanings expressed in the other perceptual modes (gaze, head and hand movement). Their system identifies actions and attentional objects (thus the semantics/meanings of the actions) in non-linguistic modes in 90.2% of the possible cases. Of all possible word-meaning pairs they recall 82.6% of them, and over those recalled pairs achieve an accuracy of 87.9% for correctly pairing words with their associated meanings. The word-like units their method extracts have boundaries that are word-level correct 69.6% of the time. In general the phone-level recognition rate is 75% correct; however, because they do not attempt to update the system's vocabulary, they don't report specific phone-error rates for their tests.

4.3 The MNVR Approach

In our MNVR technique, when a user at the whiteboard speaks an OOV label name for a chart constituent, while also writing that label name on a task-line of the Gantt chart, the OOV speech is combined with letter sequences hypothesized by the handwriting recognizer to yield an orthography, pronunciation and semantics (OPS-tuple) for the new label (Figure 4.4). The best scoring OPS-tuple, determined through a score combination technique similar to mutual disambiguation [135], was then enrolled dynamically in the system to become immediately available for future recognition.

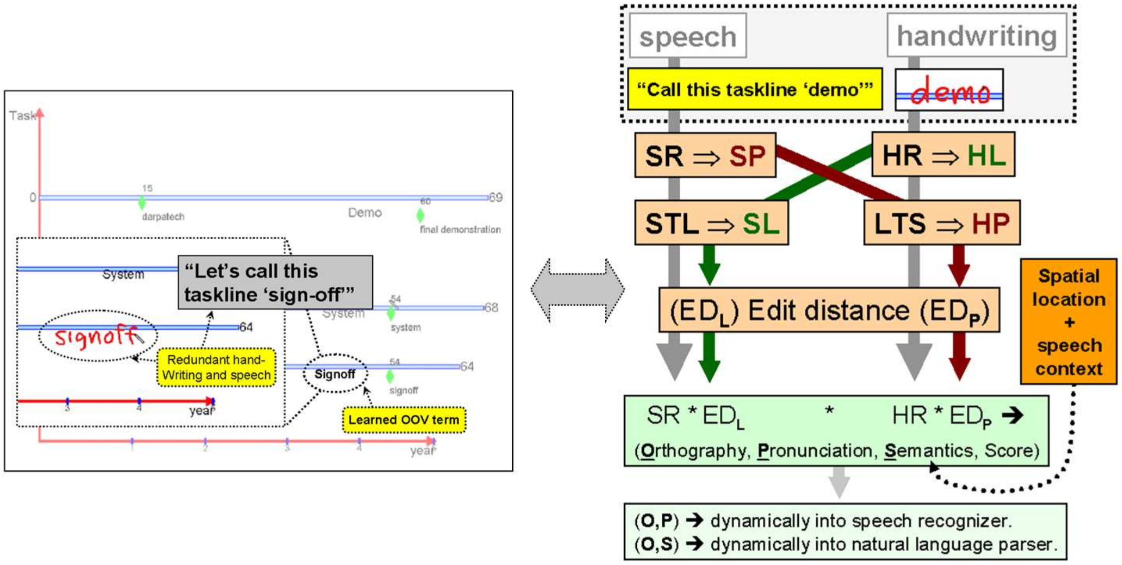

Task
0
15
darpatech
Demo
60
final demonstration
69
System
“Let’s call this taskline ‘sign-off’”
54
System
system
68
signoff
64
Redundant hand-Writing and speech
Signoff
54
signoff
64
3
4
year
Learned OOV term
1
2
3
4
year
speech
handwriting
“Call this taskline ‘demo’”
demo
SR ⇒ SP
HR ⇒ HL
STL ⇒ SL
LTS ⇒ HP
(ED_L) Edit distance (ED_P)
Spatial location + speech context
SR * ED_L * HR * ED_P ➔
(Orthography, Pronunciation, Semantics, Score)
(O,P) ➔ dynamically into speech recognizer.
(O,S) ➔ dynamically into natural language parser.

Figure 4.4: The MNVR (Multimodal New Vocabulary Recognition) technique for out-of-Vocabulary (OOV) recognition and system enrollment, via multimodal handwriting and speech. The OOV handwritten and spoken label, *signoff*, shown on the left is recognized by the process diagrammed on the right.

For example, when a user, creating a schedule chart at the whiteboard, says, “Call this task-line *handoff*,” where *handoff* is an out-of-vocabulary term, while also writing *handoff* on the whiteboard chart to label a task-line, the correct spelling (as the user wrote it) was *handoff*, but the handwriting recognizer reported the spelling to be *handifi*. Using letter-to-sound (LTS) rules on *handifi* yielded the pronunciation string, “hh ae n d iy f iy,” which was one substitution and one insertion away from the correct pronunciation of, “hh ae n d ao f." In this case the best pronunciation alternative from the speech recognizer was, "hh ae n d ao f," which was the correct pronunciation. So by using the phone string generated by the speech recognizer we were able to enroll the correct pronunciation despite errors in the handwriting recognition. The temporal boundaries of the speech recognizer's best pronunciation, "hh ae n d ao f," were determined by the second pass lattice search of the recognizer when it assigned the temporal boundary of the switch out-of in-vocabulary (IV) carrier-phrase words into the OOV segment. In this example, the end time was determined by the end of the utterance.

4.3.1 Mutual Disambiguation

Because the handwriting, speech and application modules are imperfect recognizers uncertainty is a major concern. In our previous work on handling uncertainty in multimodal interfaces [84] we have illustrated the importance of mutual disambiguation (MD). MD uses grammatical constraints to derive the best joint interpretation by unification of grammatically licensed meaning fragments across the ranked inputs of the various modes. For example, in a map-based emergency-planning scenario, sketched circles may be grammatically defined to demarcate various kinds of areas, while the specific type of area is spoken. So, one draws a circle and says, "smoke jumper drop zone," or, "sand bag area," etc. Both recognizers are uncertain and return lists of alternate results. All the grammatically licensed cross-list pairs are extracted and the highest scored one is then further processed by the system. In MNVR we have constituent lists of handwriting-derived and speech-derived phone sequences. Instead of using grammatical constraints, as MD does, to highlight the most plausible combinations within the cross-product of the two lists in the MNVR case they are all allowed. If the list of the handwriting alternatives and the list of speech alternatives are redundant then they refer to the same event. So there is no ranking to determine which of a set of grammatical rules is most likely. Rather the task is to determine which OPS tuple is most likely, as they are all grammatically licensed. MNVR uses a simple edit-distance measure, based on the spelling of phone names as the basis for re-scoring and re-ordering that cross-product list. As our results below show, phonetic-level handwriting and speech information are also capable of significantly disambiguating each other, particularly in a constrained task domain like the creation of a Gantt scheduling chart, where the temporal/spatial ontology of the task itself offers clear indications of the user's semantic intent for a given set of handwriting and speech inputs. For example, in the current implementation the creation of a schedule grid must precede the creation of task-lines, which in turn must precede the creation of task milestones. This helps the system to decide what type of constituent is being labeled, because for example if no tasklines have be recognized then the label cannot be for either a taskline or milestone.

4.3.2 System Description

In the MNVR system, users layout a schedule grid using the system's sketch-recognition agent named Charter (Figure 4.5). It employs a 2D sketch recognizer for recognizing chart constituents and a handwriting recognizer for recognizing writing [2].

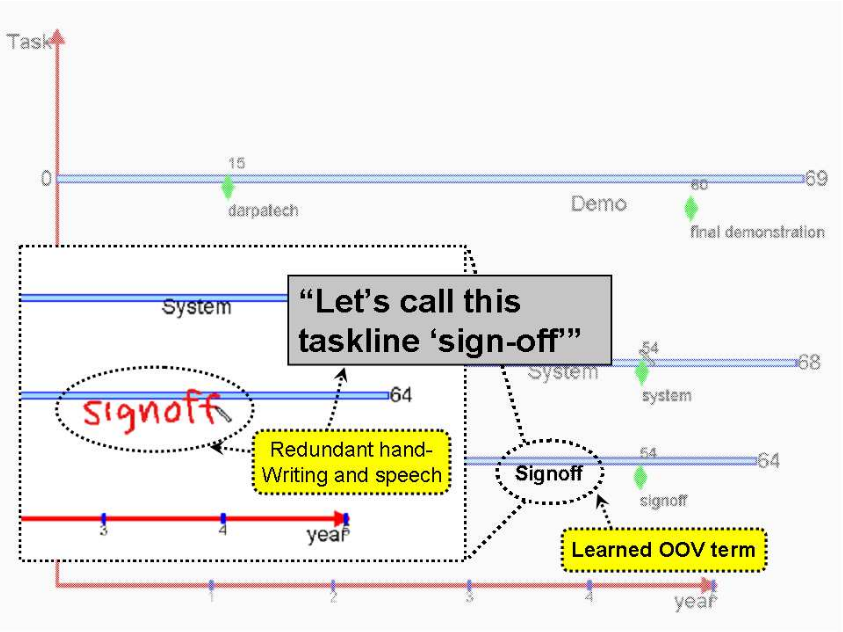


Figure 4.5: The Charter interface illustrating a beautified Gantt chart derived from sketch, handwriting and MNVR recognition. An example of MNVR recognition for the OOV label, *signoff*, is shown.

[2] The handwriting recognizer used in MNVR experiments was Calligrapher, version 5.

All modules of the system operate within an agent-based architecture. At the core of multimodal processing is an integration agent, referred to as the multiparser. The multiparser is a temporal chart parser that receives time-stamped messages from all connected input modules and integrates them to produce higher level constituents. The multiparser used in MNVR processing is directly descended from that of Johnston *et al.* [79], on which the Quickset system was based. The distributed agent-based architecture [46, 98] that underlies Quickset and MNVR is the application superstructure in which MNVR's recognition and feedback mechanisms are immersed. This superstructure is documented elsewhere [47, 44, 44, 45, 77]. We review it here because the same superstructure is used later by SHACER, and is critical in the decisions that SHACER makes about multimodal turn segmentation. Johnston [79] outlines the basic chart parsing algorithm as the following, where $*$ is an operator that combines two constituents according to the rules of the grammar, constituents are designated as terminal sequences from vertex to vertex, and both the vertices and constituents are linearly ordered.

$$Chart(i,j) = \bigcup chart(i,k) * chart(k,j) \tag{4.1}$$

As Johnston points out, in a multimodal context linearity is not assured, because input from different modal constituents can well be temporally overlapped. Thus he defines the basic temporal, multimodal chart parsing algorithm as:

$$\begin{gathered} multichart(X) = \bigcup multichart(Y) * multichart(Z) \\ whereX = Y \bigcup Z, Y \bigcap Z \neq \varnothing, Y \neg \varnothing, Z \neg \varnothing \end{gathered} \tag{4.2}$$

Constituent edges in a multimodal parse space cannot be identified by linear spans. Instead they are identified by unique sets of identifiers (e.g. multichart([s,1,1,0],[g,2,2,1])), each of which specify the constituent's mode of origin, recognition sequence number, position on the list of alternate recognitions, and semantic interpretation. This identification axiom maintains the critical constraint enforced by linearity that a given piece of input can only be used once in a single parse. Commands with intersecting IDs are different interpretations of the same input, and are thus ruled out by the non-intersection constraint in equation (2) above. This means that there can only be one correct interpretation acted upon for each set of inputs. This describes the difference between a single modal linear chart parser and a multimodal temporal chart parser.

4.3.3 Two-Level Recursive Transition Network Grammar

To implement OOV speech recognition (SR) we augmented CMU's Sphinx2 speech recognizer to use an embedded Recursive Transition Network (RTN) grammar in place of a standard n-gram language model. This architecture is schematically illustrated in Figure 4.6. We explain it here because the same embedded RTN structure underlies new word recognition in SHACER. Symbolic grammar-based speech recognition is often implemented on top of a flat lexical tree, meaning the pronunciation sequence for each dictionary word is an array of phones and the dictionary itself is thus a flat list of individual arrays. In contrast continuous speech recognizers typically encode the system's dictionary as a single phonetic *trie* structure for all dictionary words, meaning that the dictionary is not a flat list of arrays. A *trie* structure is a much more space efficient way to structure a large list of morphologically varying words than a flat list of arrays. A *re-entrant trie* structure is one that supports more than one path-walk of the *trie* simultaneously. In our augmentation of Sphinx2 we have used a separate re-entrant lexical prefix tree for each sub-grammar within the RTN (shown as terms *1-n* in each grammar in Figure 4.6). Thus when we dynamically add new words they are added only to the appropriate grammar's lexical prefix tree. During the first pass Viterbi search of MNVR's speech recognizer, all sub-grammars are searched in parallel, constrained only by the *a priori* state transitions specified in the compiled grammars.

The use of separate lexical tries for each grammar results in a system that differs significantly from that of a more standard OOV recognition approach like that of Bazzi and Glass [20, 19]. Bazzi and Glass compose separate word and phone lexicons into a single Finite-State-Transducer (FST). That composed FST has a single lexical trie containing both words and phones. This is the standard large vocabulary recognition approach. Thus, during the Viterbi search, both word and phone recognitions compete with each other in ways that do not happen in MNVR's parallel pass over separate lexical tries. For example, instead of a single language model including both words and phones, in MNVR phone

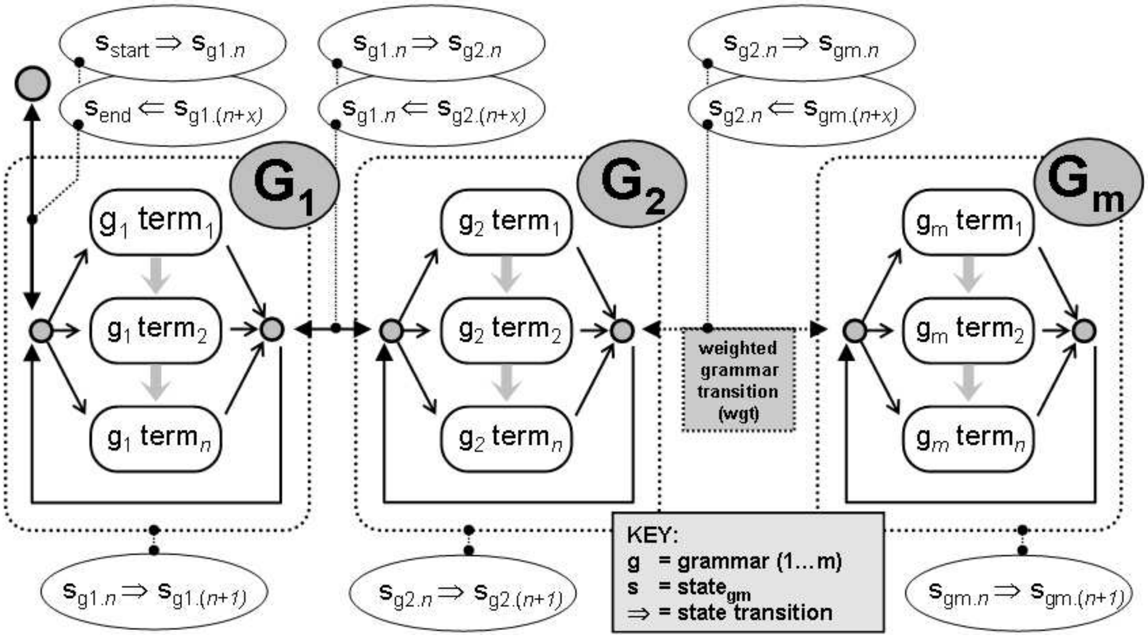

$s_{start} \Rightarrow s_{g1.n}$
$s_{end} \Leftarrow s_{g1.(n+x)}$
$s_{g1.n} \Rightarrow s_{g2.n}$
$s_{g1.n} \Leftarrow s_{g2.(n+x)}$
$s_{g2.n} \Rightarrow s_{gm.n}$
$s_{g2.n} \Leftarrow s_{gm.(n+x)}$
G_1
G_2
G_m
g_1 term$_1$
g_1 term$_2$
g_1 term$_n$
g_2 term$_1$
g_2 term$_2$
g_2 term$_n$
g_m term$_1$
g_m term$_2$
g_m term$_n$
weighted grammar transition (wgt)
KEY:
g = grammar (1...m)
s = state$_{gm}$
⇒ = state transition
$s_{g1.n} \Rightarrow s_{g1.(n+1)}$
$s_{g2.n} \Rightarrow s_{g2.(n+1)}$
$s_{gm.n} \Rightarrow s_{gm.(n+1)}$

Figure 4.6: Schematic diagram of Recursive Transition Network (RTN) architecture embedded within Sphinx2 Speech Recognizer, for the support of Multimodal New Vocabulary Recognition (MNVR).

sequence transitions are constrained by the phone sequence grammar and word sequence transitions are constrained by the word sequence grammar. Language model statistics are critical to the approach used by Bazzi and Glass, while the MNVR approach requires no statistical language model. Confining recognition to a small lexical trie for each grammar makes acoustic decisions more pronounced and easier for the recognizer. This helps to compensate for the lack of a statistical language model. Both the recognition results and applicability of these two approaches differ. For example, the fact of our not needing to create a statistical language model makes the MNVR approach much less expensive. In our case we did not have the resources to build appropriate large-scale statistical language models, so instead we created a simpler system that could still produce useful results.

In our MNVR approach the RTN is only two levels deep: (1) a task grammar, and (2) a syllabic sequence grammar to cover OOV words. Therefore conceptually our implementation is very similar to Bazzi's approach [19]. Where he uses a class-based n-gram model — with OOV terms being modeled as a single word-class — we use an RTN with an

OOV sub-grammar. The key point of penalizing the transition into the OOV component (e.g. the *wgt* in Figure 4.6) is conceptually the same for our MNVR implementation as for Bazzi's approach. Of course, the actual weight values and implementation details differ. Currently this weight for MNVR is determined empirically by repeated experimentation. To automate the determination of this weight much more data and time for testing would be required, as there are many parameterized thresholds within Sphinx 2 that have subtle and difficult to predict interactions. The basic formula for speech recognition, $P(W|A) = argmax\ (W \in L)\ P(A|W)*P(W)$, (where A = acoustic features, L = language, W = word(s), $P(A|W)$ = acoustic model, $P(W)$ = language model) is unchanged except that for MNVR the language model value at any point in the search is either 1 or 0 depending on whether the hypothesized state transition is licensed in the RTN.

The grammar writer can semantically label specific contextual locations in the grammar where out-of-vocabulary words are licensed to occur (Figure 4.7). To use the system a speaker must use these grammatically defined sentences to teach the system a new word. These sentences are called carrier phrases. Users can say other sentences while using the system but they will not be recognized. Like any grammar-based recognition system, non-grammatical speech is filtered out. In effect the system becomes a phrase-spotter for carrier phrases. The longer the carrier phrases are, the more easily they are spotted by the recognizer. At run-time, when carrier phrases occur in the speech input, their embedded OOV words are recognized (speech-recognition, **SR**, Figure 4.4) as sequences of phones (speech-phones, **SP**, Figure 4.4). This recognition of speech-phones uses the syllabic sub-grammar. The syllabifications used in that grammar are taken from the Carnegie Mellon University (CMU) Dictionary, version 6. In the following, each step's abbreviation, as used in Figure 4.4, is given in parentheses. These phone sequences are then mapped to orthographies using a sound-to-letter (**STL**) module. The orthographies are referred to as speech-letters (**SL**). If semantically interpretable handwriting recognition (**HR**) occurs co-temporally then the letter string hypotheses from the handwriting recognizer (handwriting-letters, **HL**) are mapped to corresponding phone strings (handwriting-phones, **HP**) by an embedded letter-to-sound (**LTS**) module from Alan Black's FLITE distribution [22]. The speech derived pronunciations and spellings are then paired with

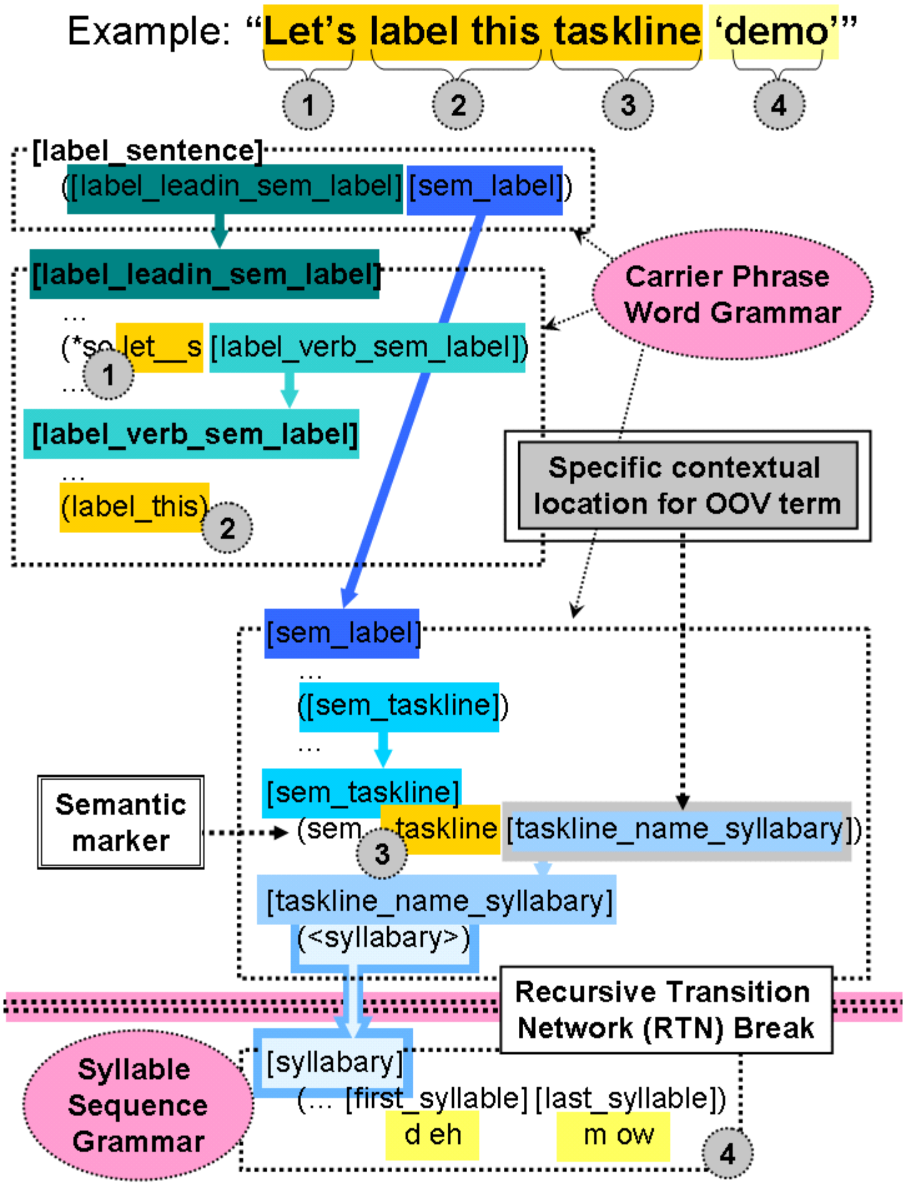

Example: "Let's label this taskline 'demo'"
1
2
3
4
[label_sentence]
([label_leadin_sem_label] [sem_label])
Carrier Phrase Word Grammar
[label_leadin_sem_label]
…
(*so let__s [label_verb_sem_label])
1
…
[label_verb_sem_label]
…
(label_this)
2
Specific contextual location for OOV term
[sem_label]
…
([sem_taskline])
…
[sem_taskline]
Semantic marker
(sem_taskline [taskline_name_syllabary])
3
[taskline_name_syllabary]
(<syllabary>)
Recursive Transition Network (RTN) Break
Syllable Sequence Grammar
[syllabary]
(… [first_syllable] [last_syllable])
d eh
m ow
4

Figure 4.7: An example of the unfolding transitions between networks in the RTN (Recursive Transition Network) for both carrier phrase word sequences (upper part of diagram) and syllabic phone sequences (lower part of diagram), with the transition between the grammars marked as a dividing line. The grammatically specified OOV positions are where the transition from upper to lower grammar is licensed.

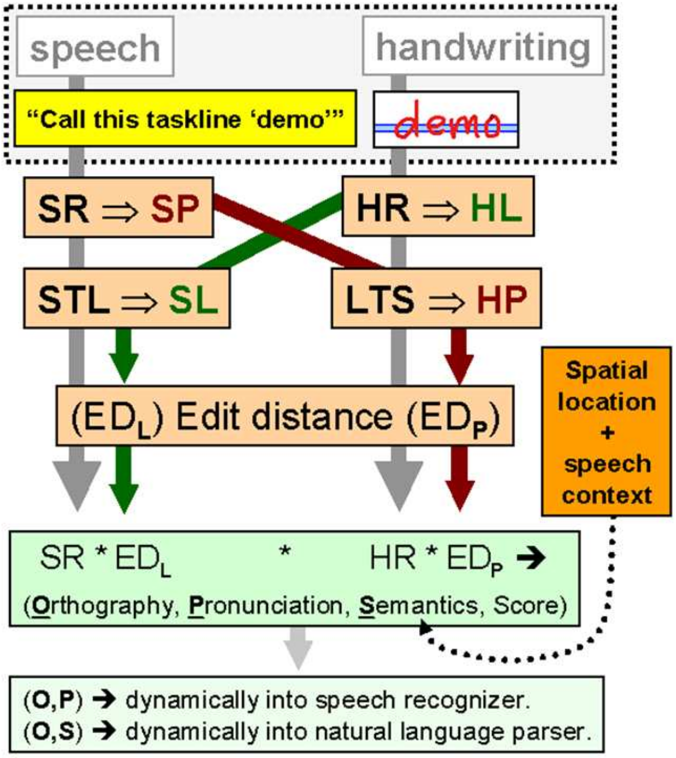

speech
handwriting
"Call this taskline 'demo'"
demo
SR ⇒ SP
HR ⇒ HL
STL ⇒ SL
LTS ⇒ HP
(ED_L) Edit distance (ED_P)
Spatial location + speech context
SR * ED_L * HR * ED_P ➔
(Orthography, Pronunciation, Semantics, Score)
(O,P) ➔ dynamically into speech recognizer.
(O,S) ➔ dynamically into natural language parser.

Figure 4.8: The MNVR (Multimodal New Vocabulary Recognition) technique for out-of-Vocabulary (OOV) recognition and system enrollment, via multimodal handwriting and speech.

the handwriting-derived pronunciations. These pairings are then scored, and the results are referred to as orthography-pronunciation scored tuples (**OPS**-tuples). That scoring uses a combined edit distance measure: **EDL** = edit-distance between letter strings, **EDP** = edit-distance between phone strings (Figure 4.4). The edit distance is modified to take matching as well as mismatching symbols into account, following [175]. The best scoring **OPS**-tuple (score = **SR** x **EDL** x **HR** x **EDP**) is then dynamically enrolled in the system at the points pre-specified during creation of the grammar. For example, task-line labels may be specified to act as modifiers for spoken references to milestones occurring on that task-line, like "move that *signoff* milestone to month fifteen." Such modifiers are enrolled simultaneously along with the new task-line's label name, *signoff*, as shown in Figure 4.9.

MNVR situates learning very specifically. Within the context of the common task of creating a whiteboard schedule chart in a multi-person meeting we recognize multimodal

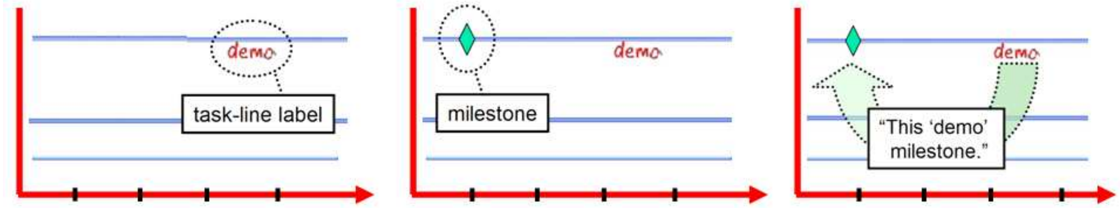


Figure 4.9: An OOV taskline label, *demo*, is shown on the left. When MNVR learns and enrolls that new label name, it is enrolled at all pre-specified spots in the grammar. Subsequently it can used to refer to the particular taskline that it labels, or to any milestone that occurs on that taskline. This is illustrated in the middle and right panels, where after enrollment a milestone placed on the *demo* taskline can be referred to as the *demo* milestone.

patterns of co-temporal handwriting, speech and spatial constraints provided by the task itself. We use the resulting notation of letter and phoneme sequences as a basis for inferring the user's intention to label a chart constituent with a new, previously out-of-vocabulary (OOV) word. This approach leverages basic perceptual qualities of Yu and Ballard's concept of *embodied intention* [176]. Two of these qualities are (1) *object-directedness* [168], which in this context means that the handwriting is written at the location where the label is meant to be, and (2) *action-connectedness* [168], which in this context means the letters of a single term are typically written from left to right one after the other with some predictable exceptions like *i* dots and *t* crosses. These qualities serve to register attentional focus and thus help in determining the semantics of their redundant multimodal labeling inputs.

4.4 EVALUATION: BASELINE PERFORMANCE TEST

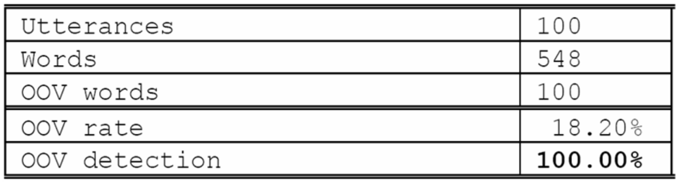
To provide baseline performance test results we collected instances of users labeling task-lines on a Gantt chart. There were three subjects. The author was one of the subjects. The other two subjects were colleagues who were not familiar with the system. One subject was female, while two were male. The labeling events involved speaking key phrases like, "Lets call this task-line *concur*," or "Label this one the *trial* task-line." *Concur* and *trial* are examples of OOV words that were recognizable within grammatically defined carrier phrases. As well as speaking, participants also co-temporally wrote the OOV label names on the task-line. There were 54 instances of such inputs collected from each subject.

Participants read their spoken input from a script. Audio was recorded using a Samson AH-1 QV wireless, close-talking microphone. Gain control was difficult to set for this microphone, so many recordings were over-modulated and unusable for speech recognition. Thus we excluded input combinations in which the speech was clipped from our test set. Some OOV terms were combinations of two short words, like *dry run* or *hand shake.* In some instances the handwriting recognizer interpreted such written inputs as two separate inputs, especially when the user paused between writing each word. MNVR was not equipped to handle two handwriting recognitions spanning a single spoken OOV term, so such input combinations were also excluded from the current test set. After exclusions we were left with 100 combinations of co-temporal speech and handwriting for labeling Gantt chart task-lines from the three users. The 100-instance data set included 18 unique carrier phrases with 51 unique embedded OOV words.

The OOV recognizer's syllabic sub-grammar had 19006 unique syllable entries spread across four categories: (1) first-last-syllable, (2) first-syllable, (3) last-syllable, (4) middle-syllable. Since there was no large corpus of task-specific speech in the test domain on which to build a plausible n-gram model over sub-word units, a symbolic grammar was used instead. Thus there were no probabilities on either syllable sequences or rule occurrences over the non-terminal categories, as would be the case with either an n-gram model or a stochastic context free grammar model. We view this as an advantage of the MNVR approach, because in modeling OOV terms it is neither desirable to (a) model only the OOV labeled words in a corpus, nor to (b) model cross-word occurrences for OOV words only at the boundaries occurring in the corpus. Both can result in over-training [20]. We argue that for task-independence, it is better to use a large dictionary to model a more general representation of the possible sub-word unit combinations of which OOV terms may be comprised. MNVR used the CMU Dictionary, version 6.

MNVR's selection of non-terminal categories was very similar to those used by Galescu [57]; however, sub-word unit combinations were restricted to a 3-syllable length limit. This is somewhat longer than Bazzi's length limit of 3-5 phones [20], while both Chung *et al's* and Galescu's systems have built in language-model-based length biases determined by the types of OOV terms occurring in their respective corpora. MNVR's 3-syllable length limit is partly due to tractability issues that arise from not having a stochastic language model. Since MNVR's second-pass search cannot rely on term sequence statistics (from a language model) for pruning, and since its syllabic vocabulary is relatively large, it cannot tractably perform a complete backward-forward A* search. So, MNVR instead relies on a depth-first beam search with a one term look-ahead that attempts to heuristically guess the best partial paths to keep in the beam. If the search dead-ends, then it back tracks to the closest previous point where a new group of partial paths outside the previous beam limit can be found and moves forward again until either the specified number of alternatives has been found or the search space is exhausted. Transitions into the syllabic sub-grammar are weighted, similar to the approach used by Bazzi [19].

Table 4.1: MNVR OOV Speech Recognition test set statistics (scored on best-of-5 output).

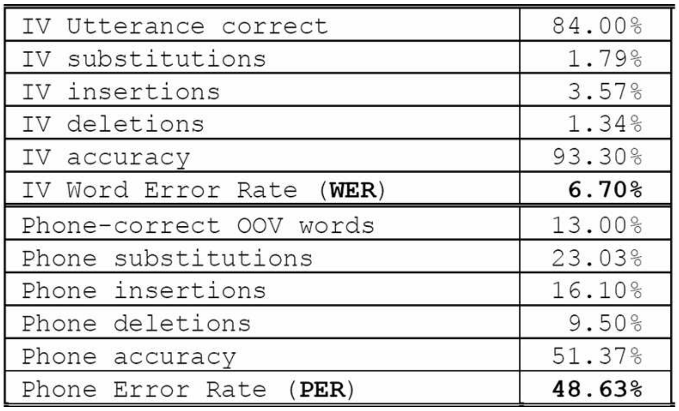


Utterances	100
Words	548
OOV words	100
OOV rate	18.20%
OOV detection	**100.00%**

The 100 test instances of multimodal speech and handwriting for labeling a Gantt chart task-line were fed into the system via the regression testing mechanism described in Kaiser and Cohen [91]. There were an average of 4.5 in-vocabulary (IV) terms in each of the 54 test instances. (Table 4.1). The OOV recognizer (OR) correctly detected the occurrence of an OOV term in all 100 instances (100% detection as shown in Table 4.1). This is the advantage of using carrier phrases. Although they are inconvenient for users to remember, they make the detection of OOV terms highly reliable.

MNVR uses carrier phrases, which are grammatically defined syntactic fragments, to frame and constrain OOV recognition to a small set of licensed linguistic contexts. The carrier phrases used simple syntax, and within carrier phrases OOV recognition slots were positioned near the end of the phrase. This was done with the fact in mind that when people are speaking didactically they naturally use intuitively simple syntax [61].

Table 4.2: MNVR Unimodal OOV Speech Recognition (scored on best-of-5 output).



IV Utterance correct	84.00%
IV substitutions	1.79%
IV insertions	3.57%
IV deletions	1.34%
IV accuracy	93.30%
IV Word Error Rate (**WER**)	**6.70%**
Phone-correct OOV words	13.00%
Phone substitutions	23.03%
Phone insertions	16.10%
Phone deletions	9.50%
Phone accuracy	51.37%
Phone Error Rate (**PER**)	**48.63%**

Our intuition was that the use of linguistic constructions used for teaching might also come naturally to people in a didactic context like a lecture or business presentation. For example, choosing to have OOV words occur near sentence final position in carrier phrases is a known characteristic of the prosodic delivery typical of infant caregivers. Certainly the 100% OOV detection rate we see in these test results bears witness to the effectiveness of leveraging sentence final position of new words to more effectively segment the phone sequences to be learned. If people in making presentations to each other use these same type of linguistic constructions, then with this approach we don't need the large number of correlated occurrences required by the associative statistical categorizers in systems like those of Roy [148] or Yu *et al.* [177]. However the assumption that people will With a single multimodal demonstration, we not only accomplish OOV detection with a high degree of accuracy, but also achieve accurate segmentation — recognizing 8.4 out of 10 of the utterances at the in-vocabulary (IV) word level completely correctly (84% Utterance correct rate, Table 4.2, line 1). So we achieve an OOV segmentation error rate of 16%. While our MNVR implementation has the ability to learn generally from a single demonstration, it will still be able to benefit from multiple presentations over time to refine pattern recognition accuracy.

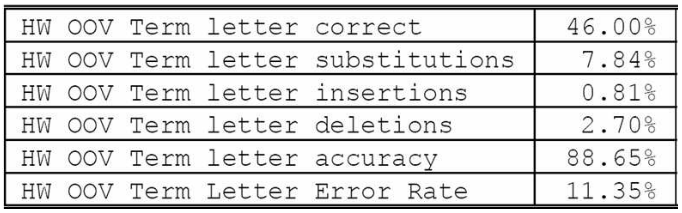
We reduced the scope of the language acquisition problem to that of recognizing out-of-vocabulary (OOV) words in grammatically specified positions. Thus, instead of posing the problem as that of language acquisition in our MNVR experimentation we have modified the problem to be additional language acquisition for an established language syntax. By using both the temporal/spatial coherence constraints of the scheduling task itself, and the contextual grammatical constraints to isolate the system's efforts at OOV recognition, we are able process new words in real-time.

The recognition rate over IV utterance words was 88.89% (Table 4.2), with 63% of the IV recognition errors being due to deletions. For example, in the utterance, "Let's label this the *handover* task-line," (in which *handover* is OOV) the word 'task-line' is deleted because the OOV recognizer doesn't find the correct boundary for stepping out of the syllable-based OOV sub-grammar in the weighted recursive-transition-network (RTN) parser embedded in the speech recognizer. Instances similar to this example account for four out of the five of the utterance level deletion errors. Adjusting the weights on the transitions between the task grammar and its embedded syllabic sub-grammar (within the RTN language model) can ameliorate this error; however, MNVR currently has no mechanism in place for dynamically adjusting this weight. Many research groups are looking at similar problems in fine-tuning word-spotting or OOV detection systems [7, 19, 116, 115]. This is a very active area of research.

Note that the IV statistics given in Table 4.1 are computed over the best five transcript alternatives produced by the recognizer. In multimodal systems it is not necessary that the best recognizer transcript be correct. Mutual disambiguation from other input modes can "pull-up" the correct transcripts [84], so we take that into account by scoring over the top five alternative transcripts. For this test set there are only two instances in which the best word-level transcript is not the recognizer's highest ranked alternative. For scoring phoneme recognition we also score over the five best alternatives from the speech recognizer, because each alternative represents a different pronunciation and only one of them has to be correct for the word to be recognized the next time it is uttered by a user. For phonetic pronunciations, the recognizer's highest ranked alternative is the best match only 48.15% of the time.

For in-vocabulary (IV) recognition, taking into account the number of substitution, insertion, and deletion errors, we achieve word-level recognition accuracy of 93.3%, and

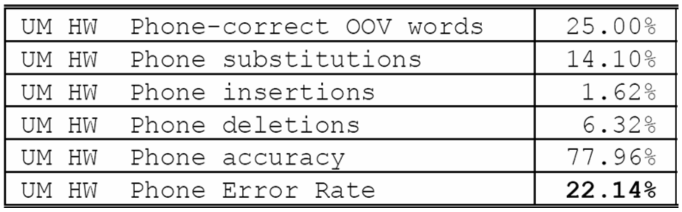
Table 4.3: MNVR Unimodal Handwriting (HW) letter recognition statistics. (Scored on best scoring handwriting alternative).



HW OOV Term letter correct	46.00%
HW OOV Term letter substitutions	7.84%
HW OOV Term letter insertions	0.81%
HW OOV Term letter deletions	2.70%
HW OOV Term letter accuracy	88.65%
HW OOV Term Letter Error Rate	11.35%

thus an IV word error rate (WER) of 6.7% (Table 4.1). The unimodal speech recognition of phonetic pronunciations is much less accurate. We achieve an accuracy of 51.37% (Table 4.2) for a phone error rate (PER) of 48.63%. Recall that Chung *et als* Speak and Spell system on a test set of 219 utterances a pronunciation-error-rate (PER) of 25.5% (much lower than our unimodal rate), and a letter-error-rate (LER) of 12.4%. Currently our MVNR systems word spelling (and thus LER) depends solely on the best alternative from the handwriting recognizer, because although there can be alternative pronunciations for the same lexical item we must still choose one single lexical representation for an item. Thus, we achieved a letter-level accuracy of 88.65% (Table 4.3) for an 11.35% LER (somewhat lower than Chungs above).

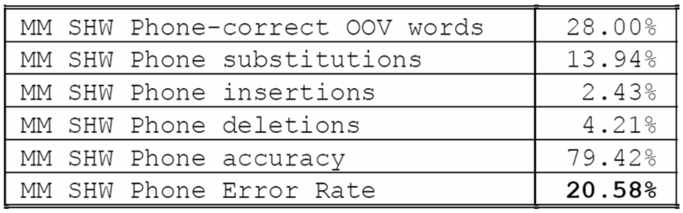
Table 4.4: MNVR Phone recognition via unimodal (UM) Handwriting (HW) using Letter-to-Sound (LTS) rules over handwriting letters. (Scored on top 5 alternatives).



UM HW Phone-correct OOV words	25.00%
UM HW Phone substitutions	14.10%
UM HW Phone insertions	1.62%
UM HW Phone deletions	6.32%
UM HW Phone accuracy	77.96%
UM HW Phone Error Rate	**22.14%**

Our unimodal PER of 48.63% is closer to that of Galescu [57] which was 31.2% — 43.2%; however, when we use LTS to generate phone sequences from the handwriting alternatives and then use these to disambiguate the speech phone sequences we improve our PER to 20.58% (Table 4.5) This surpasses the accuracy of Chung *et al's* system (25.5%), and represents a 57.5% relative error reduction between unimodal speech pronunciations

Table 4.5: MNVR Phone recognition via multimodal (MM) Speech + Handwriting (SHW) using Letter-to-Sound (LTS) rules over handwriting, and Sound-to-Letter (STL) rules over speech phone sequences. (Scored on top 5 speech and handwriting alternatives).



MM SHW Phone-correct OOV words	28.00%
MM SHW Phone substitutions	13.94%
MM SHW Phone insertions	2.43%
MM SHW Phone deletions	4.21%
MM SHW Phone accuracy	79.42%
MM SHW Phone Error Rate	**20.58%**

and multimodal speech plus handwriting pronunciations.

Of course, given such a large improvement in pronunciation recognition from unimodal speech to multimodal speech plus handwriting, we must ask how much of this improvement we could achieve solely by deriving pronunciations from the handwritten spellings transformed via LTS rules. It may be the case that speech-only information is simply not accurate enough, and we would be better off extracting pronunciations just from the handwriting. This certainly seems plausible when we recall that for this test set the letter-level accuracy of handwriting recognition is 88.65% (Table 4.3). Table 4.4 shows that using handwriting alone (with LTS transformations) we could achieve an accuracy of 77.96% in predicting the phonemic pronunciations — for a PER of 22.14%. However, when we again look at the results of combining speech and handwriting streams to arrive at pronunciations, where the PER is 20.58% (Table 4.5), we find that mutual disambiguation across multiple input modes (i.e. using speech in addition to handwriting) still yields 7.04% relative error reduction compared to extracting pronunciations unimodally from handwriting alone. This phone-level recognition improvement due to mutual disambiguation across combined speech and handwriting inputs compared to the phone-level pronunciations generated from unimodal handwriting alone is significant by a McNemar test, which yields a probability of this difference in recognition results having occurred due to chance as only 3.1e-8.

To see how using the speech-generated pronunciations helps us to improve on the handwriting generated pronunciations, we will again analyze the *handoff* example touched upon earlier. The user says, “Call this task-line *handoff*,” (in which *handoff* is OOV) while writing *handoff* on the whiteboard chart to label a task-line. The correct spelling (as the user wrote it) is *handoff*, but the handwriting recognizer reports the spelling to be *handifi*. Using LTS rules on *handifi* yields the pronunciation string, “hh ae n d iy f iy,” which is one substitution and one insertion away from the correct pronunciation of, “hh ae n d ao f.” In this case the best pronunciation alternative from the speech recognizer is, “hh ae n d ao f,” which is the correct pronunciation. So by using the phone string generated by the speech recognizer we are able to enroll the correct pronunciation despite errors in the handwriting recognition, thus demonstrating the effectiveness of using multimodal speech and handwriting to achieve a level of pronunciation modeling accuracy for new (OOV) words not achievable by either mode alone.

4.5 CONCLUSION

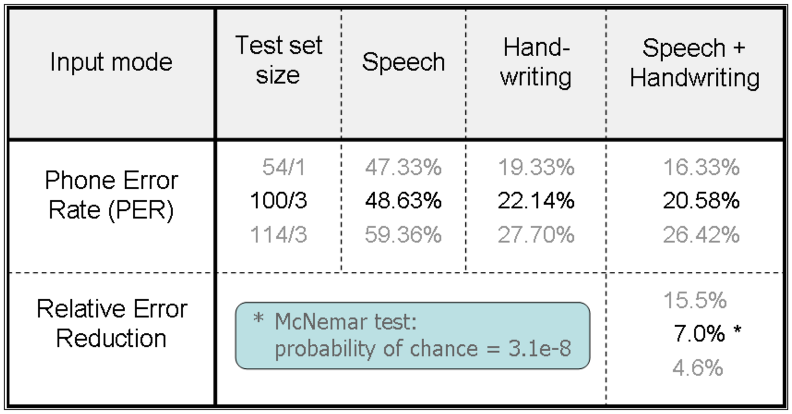
4.5.1 Robotic and Intelligent System Learning From Demonstration

Several researchers working with intelligent robotic systems in the area of learning from demonstration by vision or sensor-based motion tracking [164, 169, 160, 9] call for the inclusion of speech as a simultaneous input mode to help with the recognition of the user’s goals. There is even a growing call from researchers in AI to adopt, as a grand challenge, the learning of language from perceptual context [114]. Without context the meaning of either language or actions is often ambiguous. With context a user’s intent can be understood. MNVR uses the temporal convergence of handwriting and redundant speech as well as the spatial convergence of handwriting and chart structures as context for understanding Gantt chart labels.

4.5.2 Discussion of MNVR Test Results

The MNVR system is capable of multimodal speech and handwriting recognition. We have described a test environment where speech and handwriting in combination are used to label elements of a whiteboard chart (e.g. task-lines). Over a small test set of 100 speech and handwriting events collected from three users we have shown that combining speech and handwriting information multimodally results in significantly greater accuracy than that achievable in either mode alone. For example, the phone-error-rate over phone sequence pronunciations generated by speech alone was 48.63%, by handwriting alone it was 22.14%, while by multimodal combination of speech plus handwriting it was 20.58%. That represents a 57.5% relative error reduction compared to speech-only pronunciations, and a 7.04% relative error reduction compared to handwriting-only pronunciations (generated by LTS rules). This 7.04% error reduction is significant by McNemar test (probability of chance occurrence < 3.1e-8) as shown in Table 4.6. This significant reduction in error rate supports our hypothesis that handwriting and speech are capable of significantly disambiguating each other in a constrained task domain like that of labeling whiteboard Gantt chart constituents.

Table 4.6: Summary result for MNVR tests. The statistics reported in this section are in the middle (in black font), while the related statistics of a smaller earlier test and of a somewhat larger follow-up test are shown in gray font. In the *Test set size* column the second number represents the number of test subjects (e.g. 100/3 means 100 test instances from 3 subjects). Note that in all the tests the author was one of the listed subjects.



Input mode	Test set size	Speech	Handwriting	Speech + Handwriting
Phone Error Rate (PER)	54/1 100/3 114/3	47.33% 48.63% 59.36%	19.33% 22.14% 27.70%	16.33% 20.58% 26.42%
Relative Error Reduction	* McNemar test: probability of chance = 3.1e-8			15.5% 7.0% * 4.6%

Table 4.6 also contains related results from (1) a smaller previous study in which we found a relative error reduction for the multimodal combination of speech and handwriting compared to handwriting alone was 15.5%, and (2) a slightly larger and subsequent follow-up study on another test set of 114 labeling events in which we found a 4.6% relative error reduction for the multimodal combination of speech and handwriting compared to handwriting alone. The findings in the smaller earlier test and in the subsequent follow-up

test trend in the same direction but were not significant. It is interesting to note that as recognition in the individual modes improves, as it does here from the third to the first tests, so too does the benefit of combining information from the two streams. This suggests that as individual recognizers are improved (particularly as the speech recognition is improved) we can potentially see even more benefit by combining information from the redundant input streams

MNVR demonstrates the base-line capability of using multimodal speech and handwriting for new (OOV) word recognition. This capability allows users to teach the system their chosen vocabulary, thus shifting the burden of learning off the user and onto the system. MNVR, however, constrains users to only utter OOV terms in certain grammatically specified positions within a larger carrier phrase. For instance in the example above the carrier phrase was, "Call this task-line <*oov_term*>", and the <*oov_term*> could only be recognized in that specified position. The advantage of this approach is accuracy and tractability: MNVR is a real-time method, and the carrier phrase aids in accurate segmentation of the OOV term within the larger utterance. For some applications with fixed vocabularies (e.g. certain classes of military applications) this may be a viable approach; but, in general requiring the use of carrier phrases is too restrictive and too difficult for users to remember. A more general approach is called for, and that has motivated our development of SHACER.

4.6 SUMMARY

In this chapter we argued that learning systems in addition to their off-line-trained recognizers will need to perceive and learn from previously unseen natural demonstrations. To do this they will need to develop methods of bootstrapped learning that can assemble the outputs of their recognizers — which have been trained with supervision off-line — into meaningful higher-level symbols in real-time in response to natural demonstrations. We reviewed systems that are currently attempting to achieve such learning through the perception and grouping of related inputs that share the same relation to some object or action, like Roy's CELL and Yu and Ballard's system for understanding embodied intention.

We introduced our Multimodal New Vocabulary Recognition (MNVR) system, which leverages multimodal redundancy as the basis for learning. We distinguished MNVR from other multimodal learning systems as being capable of single instance learning, as opposed to requiring repeated inputs. Our test results for MNVR show significant reductions in error rate by integrating redundant speech and handwriting to understand chart labels, as opposed to the recognition rates of either mode alone.

MNVR is a prelude to SHACER. It shares with SHACER the perception and leveraging of multimodal redundancy for dynamically learning new words. MNVR shows that integrating information from redundant modes results in better recognition than is possible in either mode alone. However, whereas MNVR requires users to introduce OOV terms within grammatically specified carrier phrases, SHACER has no such constraint. SHACER users do not need to use carrier phrases, but can speak as they would naturally. This much less constrained language that is accommodated by SHACER brings up the need for aligning handwriting to speech in order to detect multimodal redundancy. Alignment in MNVR was simplified because the OOV positions within carrier phrases were known. Thus it was acceptable to measure phonetic distance in MNVR with a standard Levenshtein edit distance. In SHACER, we use a more sophisticated articulatory-feature based distance metric. Also in SHACER, because the phone recognition problem is much less constrained the for MNVR, we use an ensemble of variously constrained phone recognizers to completely characterize the input speech. In the next chapter we will describe these differences in SHACER in more detail.

Chapter 5

SHACER: Alignment and Refinement

The previous chapter showed that combining information from handwriting and redundant speech yielded significantly better recognition of OOV term pronunciations than was possible using either mode alone. The drawback to the technique presented in the previous chapter was that OOV terms had to be spoken in only certain grammatically defined slots within carrier phrases. In this chapter we return to a discussion of SHACER. SHACER has no constraint that users remember and speak defined carrier phrases. It can be deployed in any situation where people redundantly say what they handwrite. SHACER's aim is to unobtrusively observe human-human interactions and leverage multimodal redundancy to learn new words dynamically in context.

Figure 5.1 depicts a high-level flow diagram of input processing within SHACER. Perceptually SHACER observes participant speech and sketching/handwriting inputs. The speech is segmented into utterances separated by areas of non-speech audio. The inking input is divided into sketching and handwriting segments. The spoken utterances then flow into a set of speech recognizers: (1) a large-vocabulary continuous speech recognizer, which is referred to as the transcribing recognizer because it transforms audio input into textual transcriptions, (2) an ensemble of phoneme recognizers, and (3) a word/phrase-spotting recognizer. Each phone recognizer transforms the audio input into a sequence of phonemes. The set of those phone recognition outputs are routed to the multiparser. The results of sketch and handwriting recognition from the ink input are also routed to the multiparser. The multiparser applies temporal constraints to filter the possible combinations of phone set sequences and ink recognitions, which in turn form the constituents of rules that define how such inputs combine to layout and label a Gantt schedule chart.

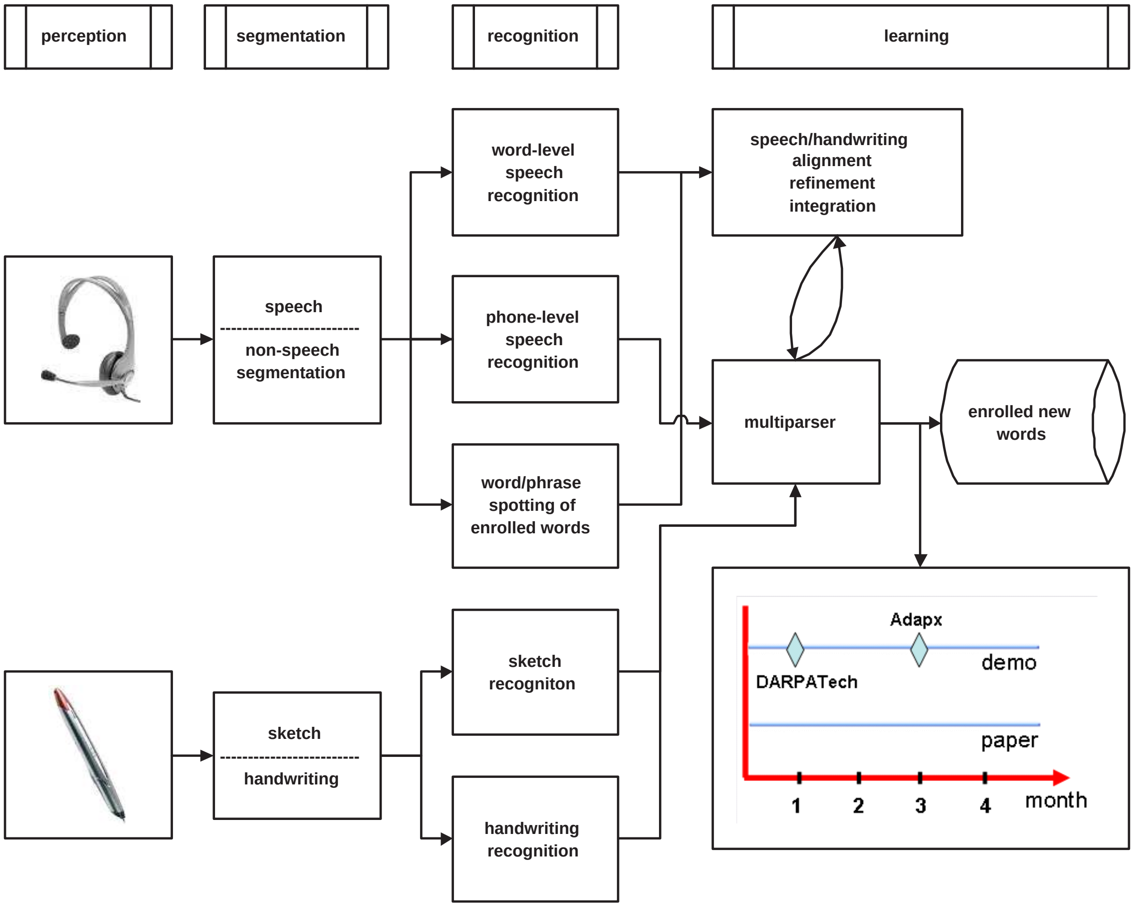


Figure 5.1: A high-level flow chart for processing within SHACER.

When phone sequences and handwriting recognitions are combined into a possible Gantt chart label, that combination is routed to a speech processing module, which performs the three main steps of SHACER processing — (1) alignment, (2) refinement, and (3) integration (as described in Chapter 3). That module returns a ranked list of possible label constituents based on detecting and combining redundant handwritten and spoken information from the phones sequences and handwriting recognition alternatives that were routed to it. These possible constituents are further processed in multiparser. If they succeed in forming a new label for the chart, then that label is routed both to the display module and to the dictionary and language model of the word/phrase-spotting recognizer. If the new label term does not already exist the word/phrase-spotter's vocabulary, then its enrollment represents dynamic learning. Once a new word is enrolled it can be more readily recognized when spoken or written again.

This chapter will explain the processes represented in Figure 5.1 in more detail. The following aspects of SHACER processing will be addressed:

1. Phone ensemble recognition.
2. Ink segmentation and recognition.
3. Alignment of handwriting and speech.
4. Refinement of handwriting/speech segment pronunciation.

The first two items preface SHACER's main processing steps, but they are important to understand. The first two of SHACER's main processing steps — *alignment* and *refinement* — will be discussed in this chapter. A discussion of the final of SHACER's three main processing steps, *integration*, will be left for the next chapter. *Integration* uses the outcome of the pronunciation *refinement* step as a decision metric for choosing the spelling and pronunciation of a new term.

5.1 PHONE ENSEMBLE RECOGNITION

In order to leverage the occurrence of multimodal redundancy SHACER must first detect it. This is conceptually illustrated in Figure 5.2. Often the redundantly spoken words that accompany a handwriting event are embedded in a longer utterance, as shown in the two examples of Figure 5.2. SHACER's approach to detecting redundancy is to align the handwriting and speech recognition outputs, and look for closely matching segments.

5.1.1 Phone Recognition in Spoken Document Retrieval

SHACER's alignment-based detection of multimodal redundancy is closely related to the problem of locating query words in a database of documents during Spoken Document Retrieval (SDR). *Documents* in Information Retrieval (IR) are objects or computer files, which may have various formats. Text, images, videos, multimedia presentations, and audio recordings can all be considered *documents* in the IR sense of the word [102]. Audio

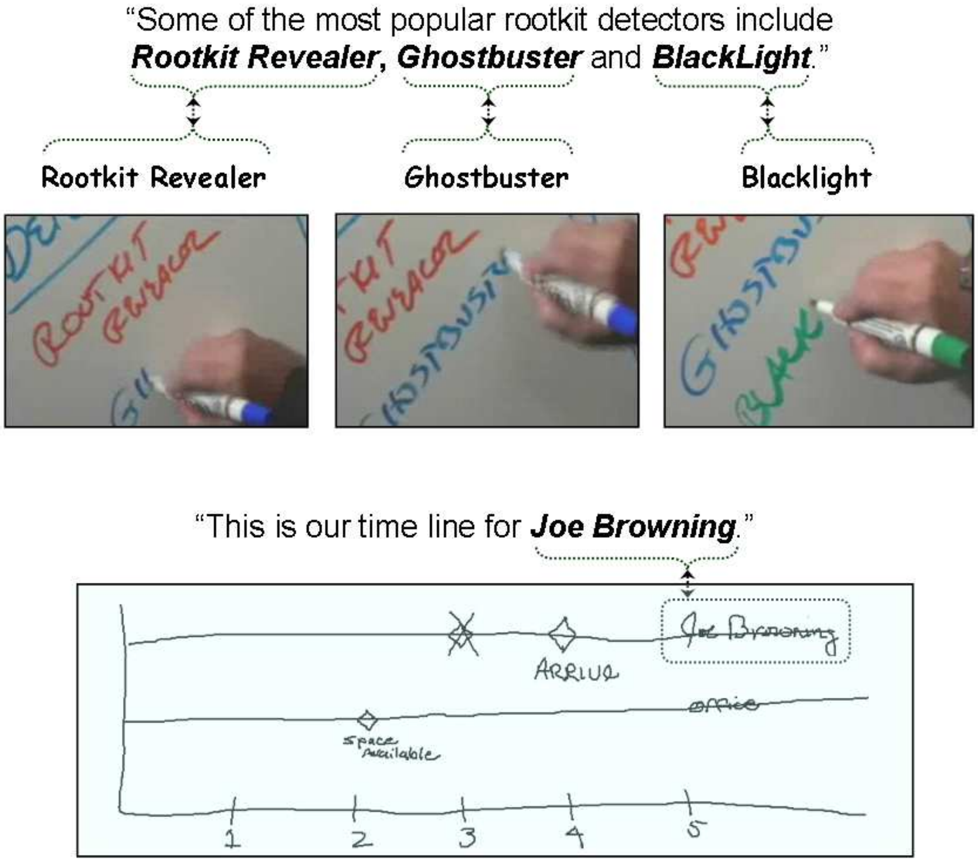

"Some of the most popular rootkit detectors include
Rootkit Revealer, **Ghostbuster** and **BlackLight**."
Rootkit Revealer
Ghostbuster
Blacklight
"This is our time line for **Joe Browning**."
Joe Browning
ARRIVe
office
space available
1
2
3
4
5

Figure 5.2: Detection of multimodal redundancy is done by finding closely matched alignments of handwriting and speech. Often times the redundantly spoken words, which accompany a presenter's handwriting, are embedded within a longer utterance. This makes the problem of aligning the two input streams and detecting the redundancy more challenging. The upper example is from a ZDNet *At the Whiteboard* presentation about Rootkits, and three products for removing them are listed. The lower example illustrates a taskline label for a new hire named *Joe Browning.* The speech in both examples is given in quotes.

recordings are referred to as Spoken Documents. Sometimes individual utterances within a longer recording are themselves considered as spoken "documents." In that case the SDR task is to retrieve all the individual spoken utterances in which a single-word search query has occurred and is relevant. This is analogous to what SHACER must accomplish in detecting the location of a spoken redundancy that accompanies a handwriting event. For example, the handwritten *Joe Browning*, in the lower part of Figure 5.2, is effectively a query term. SHACER's alignment of a handwritten term (like *Joe Browning*) with its redundant spoken occurrence in a temporally nearby utterance, is parallel to finding a typed-in query word in a database of possible audio documents during SDR.

In SDR, audio input is transformed to word or phone-level representations. These transcriptions are then organized into memory structure lists with document-occurrence counts in a process called indexation. Leath [102] and Saraclar and Sproat [153] both offer comprehensive reviews of current SDR techniques, practices and aims. For spoken documents, which are transcribed at the word-level, indexing and retrieval can basically be implemented within the standard search paradigm. Thus audio documents, represented by their automatically recognized transcriptions, can be retrieved by standard query-based web searches. This is the approach taken by the National Science Foundation's *National Gallery of the Spoken Word* project, which uses SpeechFind [183, 67] as an experimental audio index and search engine to make historically significant voice recordings freely available and accessible on the web [102]. Many other systems take this traditional transcription-based approach — like the InforMedia, SpeechBot, THISL, and NPR Online projects [102], as well as some commercial systems, like Nuance's Dragon AudioMining [127] and Virage's AudioLogger [153].

Moreau *et al.* [115] and other SDR researchers [181, 161, 80] point out that the disadvantage of using such traditional text retrieval approaches is that the search vocabulary must be known *a priori*. OOV terms, like important named entities, cannot be recognized by the system. This degrades retrieval performance. Also, the derivation of the complex language models required by the traditional transcription-based approach requires huge amounts of training data, which for constantly growing and changing audio archives may simply be prohibitively expensive to acquire and annotate. Representing audio documents as sequences of sub-word units (like phonemes) avoids this problem. Sub-word unit recognizers can be dramatically smaller than the large vocabulary continuous speech recognizers (LVCSRs) used in the traditional approach. Sub-word unit recognition is independent of vocabulary. Schone *et al.* have even proposed using sub-word unit recognition, because of its vocabulary independence, as a means of searching telephone conversations in any of the worlds languages [156]. They use 119 phones to represent all the phonemes in all of the world's languages. The disadvantage of sub-word unit indexing is that phone recognizers typically have much higher error rates than LVCSR systems. For indexing longer documents in large collections this could introduce a lose of discriminatory power, but for collections of utterance length audio documents, like voice mail or teleconference collections, this is less of a problem. For such short document databases using a vocabulary independent phone recognition system is judged by some researchers in the field to be a very reasonable approach [115]. Hybrid systems that combine word-based and phone-based recognition along with lattice-based indexation and processing are very promising [153] and have been shown to achieve better retrieval results than using word-based systems alone [181, 7, 173].

5.1.2 SHACER's Hybrid Word/Phone Recognition Approach

In SHACER handwritten words are not only likely to be OOV proper names, but because they are OOV they are also likely to be mis-recognized by both handwriting and LVCSR speech recognizers. This compounds SHACER's alignment problem. Our relying on word-level recognition alone to provide the necessary cues for detecting redundancy will not work for SHACER. Therefore, SHACER uses a hybrid approach, which combines sub-word unit recognition with word-based recognition, by aligning redundant handwriting and speech in a process that is similar to cutting-edge hybrid SDR systems. SHACER's LVCSR word-level recognizer is run in parallel with an ensemble of phone recognizers (Figure 5.1). Currently both recognizer types are run off-line. Speed has not been a focus of SHACER's development efforts for the system described in this thesis. In the future, as more resources for SHACER's development become available, it could take advantage of better, faster phone-level recognition. For example in recent research on keyword spotting in informal speech documents, phonetic processing speeds of 0.02 x real-time [161] have been reported.

Each of SHACER's four ensemble phone recognizers is constrained differently. Figure 5.3 illustrates some of the various phone sequence recognition outputs and their alignment with respect to each other. We use both phones and syllables as sub-word units. Both SDR and name recognition approaches have shown that better phone-level recognition can be achieved by using syllables as sub-word units [181, 159]. The transformation from syllables to phone sequences is trivial because we name syllables by their respective phonetic pronunciation sequences (e.g. cat = "K_AE_T" = "K AE T"). The four constraints are: (a) syllables follow a grammar of English syllabic sequencing (see Section B.2), (b) phones follow a grammar of English phone sequences (see Section B.3), (c) any syllable can follow any other with equal likelihood, and (d) any phone can follow any other with equal likelihood.

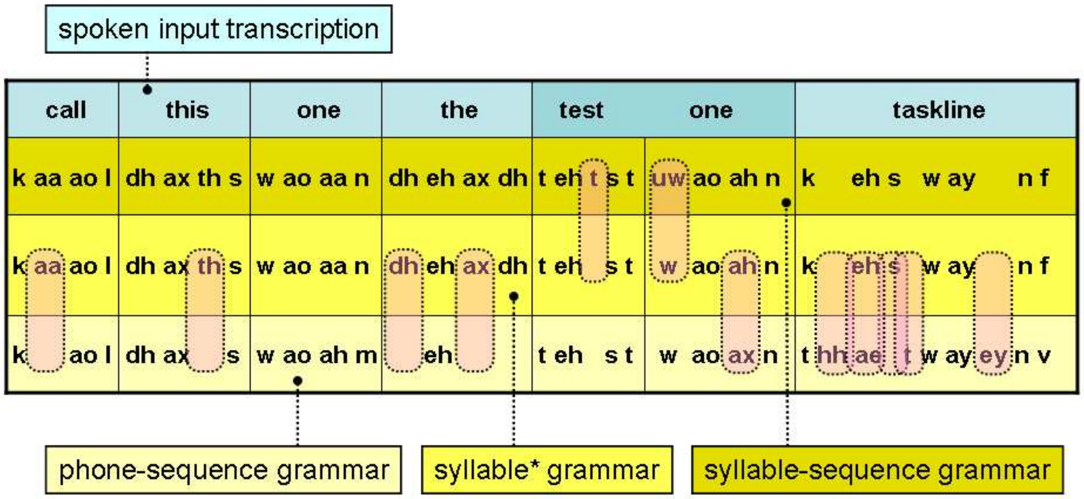

spoken input transcription
call | this | one | the | test | one | taskline
k aa ao l | dh ax th s | w ao aa n | dh eh ax dh | t eh t s t | uw ao ah n | k eh s w ay n f
k aa ao l | dh ax th s | w ao aa n | dh eh ax dh | t eh s t | w ao ah n | k eh s w ay n f
k ao l | dh ax s | w ao ah m | eh | t eh s t | w ao ax n | t hh ae t w ay ey n v
phone-sequence grammar
syllable* grammar
syllable-sequence grammar

Figure 5.3: Phone sequence outputs for different ensemble recognizers: (bottom) unconstrained phone-sequence, (middle) unconstrained syllable sequence grammar (the *, or *star*, means that any syllable can follow any other) , (top) constrained syllable sequence grammar. The differences between outputs are highlighted.

SHACER employs an ensemble approach to phone recognition for several reasons. The first reason for using an ensemble of phone recognizers is that phone recognizers have high error rates, so in an attempt to compensate for this we constrain each recognizer differently. Our hope is that by this means the correct phones and phone sequences at each position will more often be available in the ensemble alignment matrix. This is illustrated in Figure 5.3 where the frequent differences between the outputs are highlighted. The way that SHACER extracts the phonetic information from an alignment matrix is more complex than just a simple majority vote at each position. A positional bigram model of phone sequences is extracted from the alignment. This model constrains a second pass phone-level recognition, which is described in more detail later in this chapter. Thus, information from the alignment matrix is used like a language model. Both the existence of phones in the matrix and their positions relative to each other is taken into account by the sequence model. During the second pass phone recognition, information from the alignment-matrix-derived phone sequence model is weighted in relation to the phone-level acoustic scores. This weighting serves to scale the scores of the sequence model in relation to the acoustic model scores, so that scores from one model do not overwhelm the scores from the other model. In speech recognition this weight is called the language model scaling factor, and is usually determined by empirical trial. Thus, rather than a majority vote of which phones are best in which positions, SHACER uses both (a) alignment-based phone sequence information and (b) acoustical information to create a refined pronunciation. The second pass recognition, which refines the pronunciation hypothesis, will be explained in more detail later in this chapter.

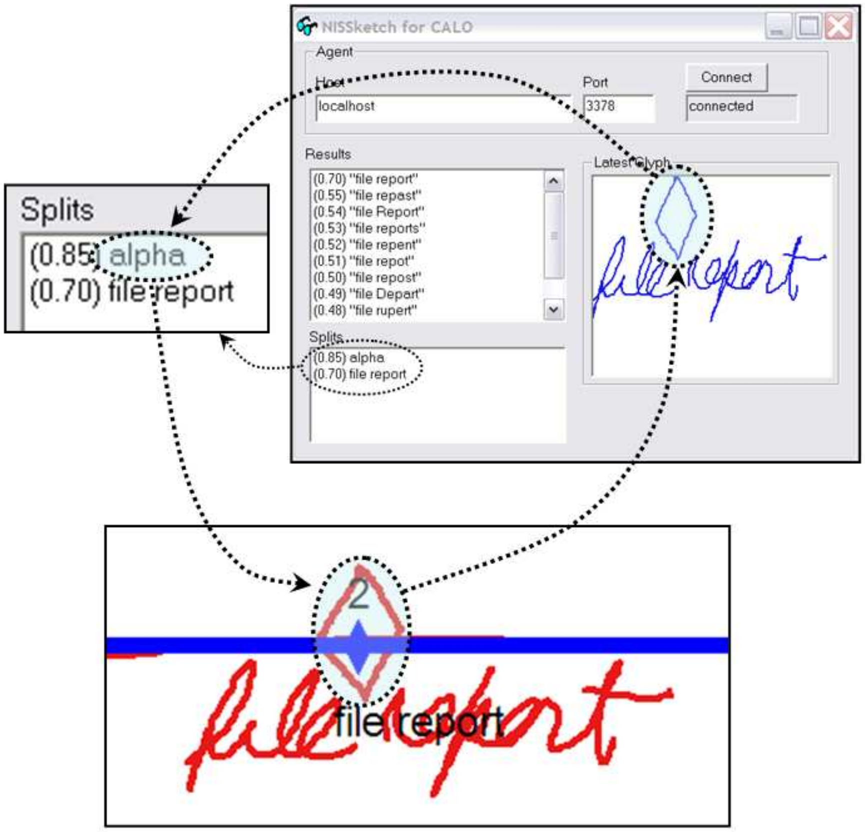
The second reason for using an ensemble of phone recognizers is that SHACER's phone recognizers are all grammar-based. They use no statistical model of English phone sequencing during first-pass phone recognition. We did not have the resources available to build or acquire such a statistical model of English phone sequences. Since SHACER does not use a stochastic model of English phone sequences, its producing a list of alternate phone sequence hypotheses tends to be intractable. Sequence models are a necessary constraint during the A* search used to produce such alternates lists [145]. If we substitute hard threshold pruning in place of using sequence models, then variations in resulting interpretations tend to be bunched toward the end. Therefore, we use an ensemble of outputs from differently constrained Viterbi, first-pass phone recognitions. This allows for fuller variation across each alternate interpretation, rather than just at the sequence ends.

The third reason for using an ensemble of phone recognizers is that individual phonetic time boundaries must be known. This is critical for SHACER's approach to locating and detecting OOV terms. Using longer sub-word units (like syllables) provides better phone-level recognition accuracy; but, within-syllable phonetic time boundaries are not easily recoverable. There are methods for recovering these phonetic boundaries. Siede *et al.* [158] replace syllable-like units with their equivalent phone sequences by an approach that is similar to techniques for automatic time-alignment of phonemes [71]. This could also be done by an augmentation of the low-level data structures used to represent states within the Hidden Markov Models (HMMs) that are traversed at the lowest levels of the recognizer's first-pass Viterbi search [174]. Instead of implementing these approaches SHACER simply uses both syllable and individual phone based recognizers in its ensemble of phone recognizers. For syllable-based phone recognizers the within-syllable phone transitions are very roughly estimated by simple interpolation with respect to the syllable start and end times. For individual phone recognizers the temporal information for phonetic boundaries is fully available. During processing SHACER heavily discounts within-syllable temporal phone boundaries and instead mostly depends on temporal boundaries from the individual phone recognizers. So ensemble recognition supports both syllable-based phone recognition for higher accuracy and phone-level recognition for better temporal boundary resolution.

In summary, SHACER uses a hybrid recognition approach that combines both word-level and phone-level recognition because, as in the SDR task, that approach facilitates OOV recognition. SHACER uses an ensemble of differently constrained phone recognizers in an effort to provide the best phone level information available, given poor phone recognizers with little-to-no phone sequence modeling. Since the goal of this thesis is to prove that SHACER can learn new words by leveraging multimodal redundancy, all that is needed is adequate rather than state-of-the-art phone-level recognition. In the future SHACER can benefit from better sub-word unit recognition.

5.2 INK SEGMENTATION

In our current implementation of SHACER we use NISSketch (a commercial product from Adapx, *http://www.adapx.com*) both for sketch recognition and as a wrapper for the MS Tablet PC Handwriting recognizer. Successful sketch and handwriting recognition are highly dependent on correctly segmented input. If sketch strokes are grouped together with handwriting strokes then sketching is likely to be interpreted as handwriting and *vice versa*.



Host: localhost
Port: 3378
connected

Results
(0.70) "file report"
(0.55) "file repast"
(0.54) "file Report"
(0.53) "file reports"
(0.52) "file repent"
(0.51) "file repot"
(0.50) "file repost"
(0.49) "file Depart"
(0.48) "file rupert"

Splits
(0.85) alpha
(0.70) file report

file report

Figure 5.4: To process input-inking SHACER needs to segment out chart constituents from handwriting, like separating out the highlighted milestone *diamond/alpha* symbol from the label-name handwritten below it. In this example the *diamond/alpha* was drawn and the *file report* label was immediately handwritten below it. There was no temporal cue as to the correct segmentation.

5.2.1 Segmentation Rules

To accomplish handwriting and sketch recognition SHACER needs to segment ink-gestural input into its component sketch and handwriting segments (Figure 5.4). The approach is to distinguish handwriting from the other sketch constituents of Gantt charts like axes, lines, milestone-diamonds, cross-outs, etc. In the following discussion an ink *stroke* is defined as a sequence of x/y screen coordinates tracing the position of a pen tip from surface contact at pen-down until the next pen-up. The features of ink strokes which are tracked are: (1) individual stroke size relative to screen size, (2) stroke closeness to the previous stroke, (3) horizontal relation to previous stroke group, (4) vertical relation to previous stroke group, (5) height/width ratio of stroke group's bounding box dimensions, and (6) temporal distance from previous stroke.

1. *Individual stroke size relative to screen size*: For SHACER's Gantt chart domain, this feature effectively filters out the large Gantt Chart axes from all other ink strokes. The axes are constructed as a single stroke that traces out a large "L" shape. The nature of Gantt charts is that the axes will typically be larger than any other sketched or handwritten constituents that become part of it. The axes roughly define the horizontal and vertical boundaries of the chart.

2. *Stroke closeness to the previous stroke*: This feature can be used to hypothesize word breaks. However, SHACER is very conservative in proposing word breaks. Instead it treats all horizontally contiguous strokes as part of the same group and lets the handwriting recognizer hypothesize word breaks within a group. Only when a stroke is horizontally *very* distant from a previous stroke is a break hypothesized and a new stoke group started.

3. *Horizontal relation to previous stroke group*: All strokes that are (a) relatively close to the previous stroke, (b) don't precede the previous stroke beyond some threshold of distance, and (c) don't either start below or extend too far below the bounding box of the previous stroke group are considered an extension of the previous stroke group. This is illustrated in the *horizontal extension* portion of the upper panel of

Figure 5.5.

4. *Vertical relation to previous stroke group*: A stroke that is below the previous stroke group starts a new group, triggering recognition on the previous group. This is illustrated at the *vertical shift* points in the upper panel of Figure 5.5.

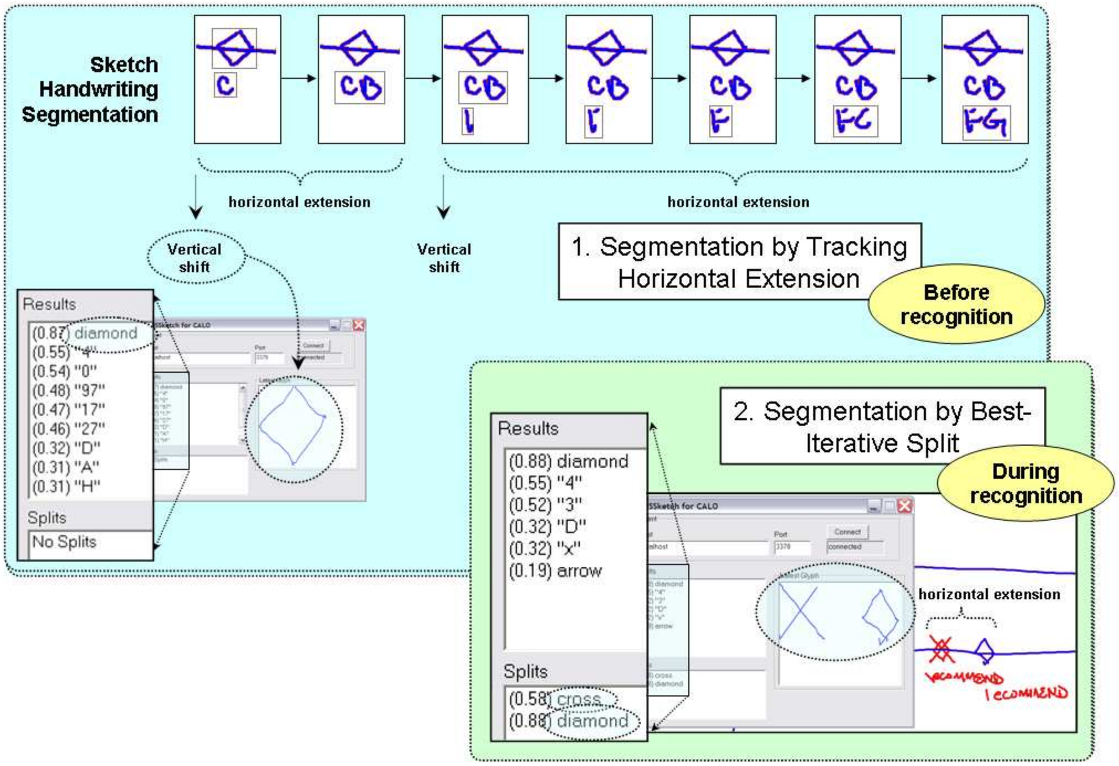
5. *Height/width ratio of stroke group's bounding box dimensions*: Groups of handwriting strokes characteristically maintain a heighth/width relationship. If the group is too compressed or extended either horizontally or vertically beyond a normal range for handwriting, then it is considered to be sketching rather than handwriting. For example, a horizontal line (which could be a taskline or cross-out stroke) will typically have a very low height to width ratio that distinguishes it from handwriting. However, sending such a stroke group to the handwriting recognizer it will typically be recognized as low-confidence handwriting.

6. *Temporal distance from previous stroke*: When the time after the previous stroke exceeds a threshold, then that triggers recognition on the previous stroke group.

Figure 5.5 illustrates how tracking vertical shifts and horizontal extensions of the bounding areas of accumulated ink strokes helps to inform the segmenter of constituent boundaries. Thus single lines of handwriting (perhaps prepended by small symbols like a cross-out or milestone diamond) can be reliably segmented. A disadvantage of this approach is that it rules out the recognition of multi-line handwriting as a single segment for recognition. An advantage is that in practice it can handle the segmentation of single lines of slightly diagonal handwriting.

5.2.2 Iterative Best-Splitting of Segmented Stroke Groups

Stroke size is a good feature for distinguishing handwriting ink from the sketch ink of axes and task-lines in SHACER's Gantt chart domain; however, there are also some smaller chart symbols (e.g. tick-marks, cross-outs, and milestone-diamond symbols) used in the Gantt charts that SHACER processes. Given this, it does happen that concatenations of non-handwriting symbols together with handwriting can occur. There can also be



Results
(0.87) diamond
(0.55) "4"
(0.54) "0"
(0.48) "97"
(0.47) "17"
(0.46) "27"
(0.32) "D"
(0.31) "A"
(0.31) "H"
Splits
No Splits

Results
(0.88) diamond
(0.55) "4"
(0.52) "3"
(0.32) "D"
(0.32) "x"
(0.19) arrow
Splits
(0.58) cross
(0.88) diamond

Figure 5.5: Sketch/Handwriting segmentation: upper-pane, by tracking horizontal extension; lower-pane, by a best-iterative split within the sketch recognizer itself.

concatenations of non-handwriting symbols to other non-handwriting symbols — like the cross-out symbol followed by milestone-diamond symbol shown in Figure 5.5's lower pane. Such a concatenation is treated as a single input segment for recognition. SHACER's sketch recognizer therefore performs an iterative best-split search on all incoming ink segments, specifically looking for instances in which the first 1-4 strokes can be better recognized as a separate cross-out or milestone-diamond symbol. This is illustrated in the lower pane of Figure 5.5 where the recognizer splits the concatenation into its constituent parts — a cross-out followed by a milestone-diamond. The same technique of best-iterative fitting can segment-out milestone-diamonds that on input are clumped together as a single segment with nearby handwriting strokes. This is an exhaustive approach, trying all possible combinations of segmentation splits. For 1-4 stroke prefixes this works, but for the general case of separating sketch symbols from handwriting this approach may be too computationally expensive.

5.2.3 Correct Segmentation Rates

These ink segmentation techniques are heuristic and specific to SHACER's Gantt chart domain. As SHACER improves these segmentation methods will need to be upgraded, expanded and brought into line with state-of-the ink segmentation approaches. For our current purpose of supporting the development of SHACER's integrative learning capability they suffice and in fact can segment the ink input from the development test series of meetings with 100% accuracy. Note that 100% segmentation does not imply 100% recognition. Segmentation just means that strokes were grouped correctly into sketch versus handwriting strokes groups, or mixed sketch/handwriting groups in which at most only the first 1-4 stokes were sketching. Even with correct segmentation it still happens that a group of handwriting strokes may be recognized as some type of sketch symbol like a line. For these segmentation rules, the development test set was treated as a training domain. Each example of incorrect segmentation was analyzed and new features were identified that could correct the segmentation error. For the entire development test set, features were added in this way until a feature configuration was reached that accomplished the desired level segmentation accuracy. Again this approach is only successful because the domain is small and limited enough that it could be approached in this way.

5.3 IDENTIFYING HANDWRITING AND SPEECH REDUNDANCY

A sub-task within SDR is to actually word-spot the individual occurrences of query terms in the audio database [180, 72, 7], as opposed to just determining which spoken documents are most relevant to the query. As pointed out above, this word-spotting task is conceptually the same task that SHACER must accomplish in aligning handwriting and speech redundancies. Handwriting in SHACER's domain can be considered the equivalent of a query term whose words must be spotted in the surrounding spoken utterances. The end-goal for word-spotting in SDR is to retrieve an audio document or segment for play back in the retrieval interface. The end-goal for word-spotting in SHACER is to dynamically learn the spelling, pronunciation and contextual semantics of a redundantly presented word and enroll it in the system's vocabulary to improve subsequent recognition. There is no dynamic learning of new words involved in SDR.

5.3.1 Word Spotting and Cross-Domain Matching in SHACER

There are two approaches to the word-spotting task within the SDR research community. One approach is the vector space model (VSM), which defines a space in which both documents and queries are represented by vectors. Each vector is composed of term/weight tuples, which can also store positional information. A typical means of assigning the weight or relevance of a term in a document is the *tf-idf (term frequency - inverse document frequency)* weighting scheme [151]. The process of creating the term/weight tables for a given database of documents is called indexing. The VSM approach can used with either transcripts or lattices [153, 185, 181]. If the query keywords are represented as words then the lattices are word-level lattices. If the query keywords are represented as phoneme sequences then the lattices are phone-level lattices. Transforming query keywords to phone sequences is done by a text-to-speech engine; or alternatively, when the query words are spoken, speech recognition automatically provides phonetic pronunciations. Presently the main current of SDR research is indexing both word and phone lattices together, so that query keywords can then be treated as words when they are in-vocabulary and treated as phone sequences when they are out-of-vocabulary [153, 173, 7, 80]. Retrieval based on VSM searching of word lattices is fast and scalable to large databases. VSM searching of phone lattices is at least an order of magnitude slower [28] than searching word lattices; however, both research systems [174] and commercial systems offer very fast searching based on phone matrices. For example, for a commercial system (e.g. Nexidia [122]) that pre-processes audio to produce searchable phonetic tracks, Cardillo *et al.* [32] report phonetic pre-processing rates of 4 times faster than realtime (4 x realtime) and search rates of 36,000 x realtime (equivalent to searching 10 hours of media recordings in 1 second for significant queries).

The second approach to word-spotting in SDR relies on dynamic programming matching techniques that don't use vector indexation [74, 75, 76]. This approach hypothesizes location slots where query words could exist in the document database, estimates the probability of a slot/query-word match using some sort of probabilistic edit distance measure, and then computes the relevance of a document based on those probabilities [115]. This approach is much slower than VSM techniques due to the computational cost of slot detection and probabilistic distance matching. However, for small databases like finding utterances in which a certain city name was spoken it performs significantly better than VSM approaches, as reported by Moreau *et al.* [115]. It also possible to use this approach to find partial matches to spoken queries, so that users may utter queries that contain extraneous words like “well,” or ”let’s see” [76]. Another use of this approach is finding repeated words in lectures or other recorded audio [141, 76]. Repeated words in lectures tend to be important, subject-specific words, so this approach could aid in the process of identifying and potentially learning new words, as reported by Park and Glass [142] in recent work at MIT.

SHACER uses a dynamic programming matching technique as opposed to a VSM technique for word-spotting redundancies across handwriting and speech. Currently SHACER does exhaustive dynamic programming (DP) searches to discover redundancies, but the window of spoken utterances that are examined is relatively small. Currently the five utterances temporally preceding the moment at which the DP search starts are examined. DP matching techniques for small databases, where speed is less of an issue, perform significantly better than vector space modeling techniques [115]. In the future we will experiment with VSM approaches for identifying the particular utterances in which redundancies are highly likely to be located, and then within those utterances deploy a DP search. SHACER’s DP search could potentially be faster by using optimization techniques like those described by Itoh [76, 75].

For discovering repeated spoken words during lectures both Itoh [76] and Park [142] match speech to speech. SDR systems that allow spoken queries, like that of Moreau *et al.* [115, 64, 124] also match speech queries to a spoken database. SHACER’s matching task is complicated by having to perform cross-domain matching from handwriting to speech. Some work on dynamic programming algorithms specifically for cross-domain matching between handwritten queries and text produced via Optical Character Recognition of scanned documents has been described by Lopresti *et al.* [104, 105]. However, we are not aware of any other system that performs cross-domain matching between handwriting and speech as SHACER does. Kurihara *et al.* have developed a system called *SpeechPen* that uses speech recognition to allow note takers to create verbatim transcripts of spoken Japanese instructions or lecture presentations. It allows users to choose from a list dynamically predicted speech recognition alternatives to extend their current note-taking strokes and thus increase the speed of taking verbatim notes. Currently *SpeechPen* does not perform any DP matching between handwriting and speech. Schimke *et al.* [155] have proposed an architecture for collecting time-stamped speech and handwriting, with an aim to integrating them for increased recognition accuracy, but have not to our knowledge reported on an actual implementation.

5.3.2 Multiparser Alignment Requests Within SHACER

The alignment process is conceptualized in Figure 5.6. The inputs to the alignment process are: (a) LVCSR speech transcripts, (b) word/phrase-spotter recognitions, (c) phone ensemble transcripts, (d) handwriting recognitions. All but (c) are ranked lists of alternate recognitions, with each alternate having a likelihood score associated with it. The phone ensemble outputs are one phone sequence hypothesis per recognizer. An example of the information in the alternates list for handwriting recognition is shown in Figure 5.7. Aside from ranking scores each alternate is also paired with its letter-to-sound (LTS) transformation. An embedded module within the multiparser acquires these LTS transformations. That embedded LTS module along with an accompanying sound-to-letter module is ported from Alan Black's CMU FLITE toolkit [22].

The multiparser's primary role is temporal grouping. It groups phone ensemble outputs together by examining their time stamps. It then examines the time stamps of incoming handwriting recognition alternates lists and proposes combinations of those phone ensemble output groups and handwriting recognition alternates lists whose time stamps are within some temporal threshold of each other. These proposed combinations are then routed to the alignment module, which is labeled as the *ARI* module in Figure 5.6. The multiparser requests that the *ARI* module test the combination for the occurrence of handwriting/speech redundancies, and return a ranked list of spelling/pronunciation/semantics

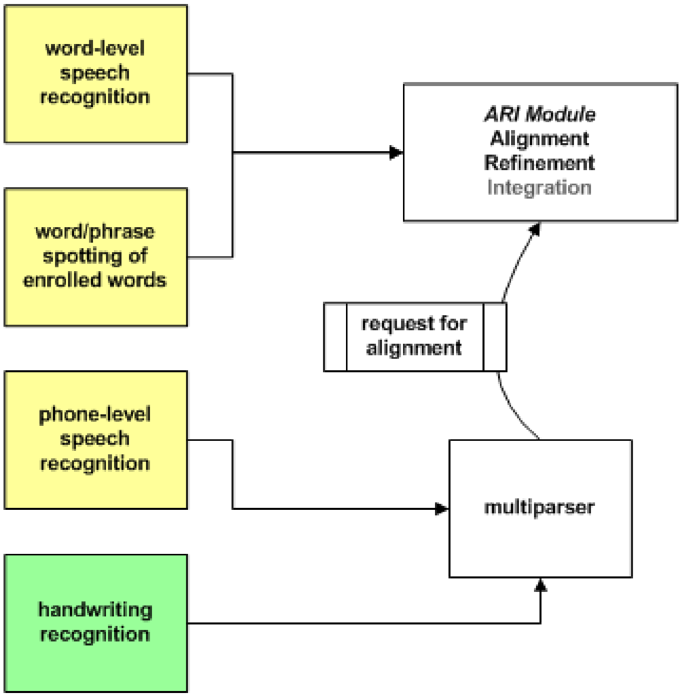
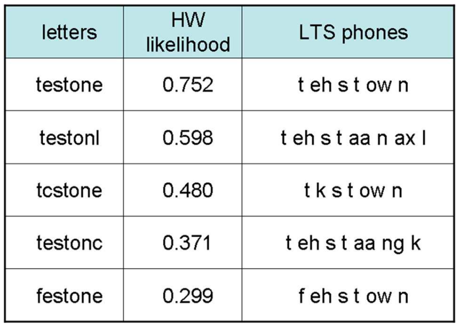

word-level speech recognition
word/phrase spotting of enrolled words
phone-level speech recognition
handwriting recognition
ARI Module
Alignment
Refinement
Integration
request for alignment
multiparser

Figure 5.6: This diagram shows a portion of SHACER's high-level flow chart (see Figure 5.1 for the entire flow chart) showing how outputs from the transcribing speech recognizer and from the word/phrase-spotting recognizer are routed to the Alignment, Refinement and Integration (ARI) module, while outputs from the ensemble of phone recognizers and from the handwriting recognizer are routed to the multiparser. The multiparser combines phone ensemble outputs with handwriting recognizer outputs and then requests that ARI perform alignment on those combinations.

tuples for any hypothesized redundant terms. In this section we only address the *ARI*'s alignment process. Its refinement process is dealt with in a later section, and its integration process is described in the next chapter.

When the ARI module receives the request for alignment from the multiparser, it attempts to activate the transcript and lattice information for the utterance associated with the phone ensemble outputs. A sliding window of previous utterance information is maintained that serves as the system's short-term memory. If the phone ensemble's utterance is within that window (currently set to the previous five utterances) then activation of transcript and lattice information is effected by reading the appropriate files into active memory structures. The primary purpose of alignment is to judge whether the handwriting was spoken redundantly within the utterance with which the multiparser has



letters	HW likelihood	LTS phones
testone	0.752	t eh s t ow n
testonl	0.598	t eh s t aa n ax l
tcstone	0.480	t k s t ow n
testonc	0.371	t eh s t aa ng k
festone	0.299	f eh s t ow n

Figure 5.7: List of handwriting (HW) recognitions for the handwritten phrase, *test one*. Each alternative has a spelling, score, and letter-to-sound (LTS) phone sequence.

paired it in the request for alignment. If it turns out that a local alignment is found that is close enough then the inputs are judged to be redundant.

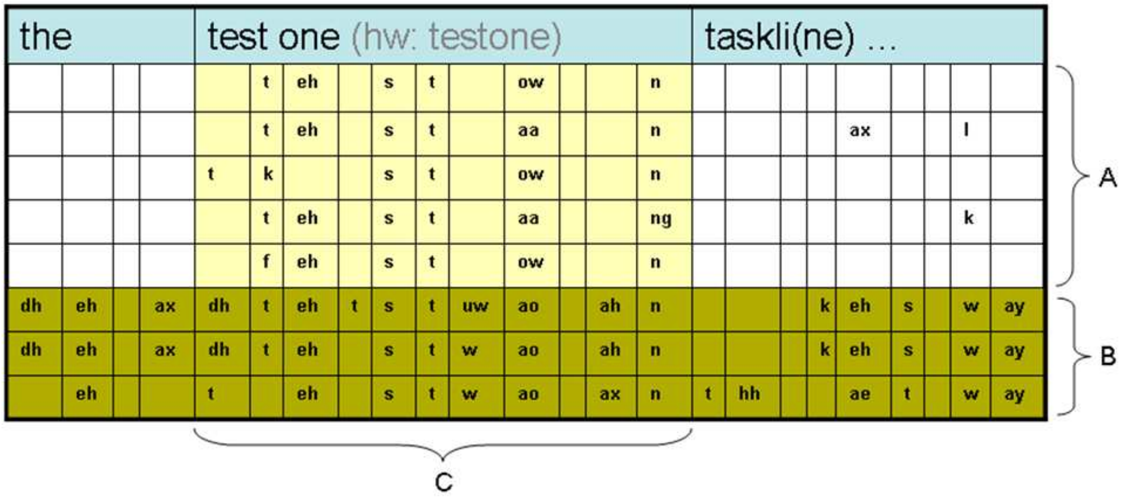
Judging whether a Phonary Request's Speech/Handwriting Are Redundantly Associated

Judging whether the speech and handwriting included in a request for alignment from the multiparser are indeed redundantly associated has several steps. The first step is to check for a transcript match of handwriting letter-string alternatives to terms in the large vocabulary continuous speech recognizer (LVCSR) transcript. If there is an exact match then the redundancy judgement is trivial and subsequent processing is reduced to exploring alternative pronunciations present in the phone ensemble outputs, which might help in dynamic pronunciation adaptation.

If there is no exact transcript match then the handwriting and speech are phonetically aligned with each other. Figure 5.8 shows an example of such an alignment.

5.3.3 Phonetic Articulatory-Features as an Alignment Distance Metric

To perform this alignment SHACER uses a phonetic articulatory-feature based alignment technique based on work by Kondrak [96, 95]. Many researchers in SDR measure phonetic distance by performing speech recognition on a training corpus, and then building a statistical model of the frequency with which one phone is mis-recognized as another phone by the recognizer [116, 7, 76]. The phone-to-phone matrix in which these statistics



the				test one (hw: testone)											taskli(ne) ...								
					t	eh		s	t		ow			n									
					t	eh		s	t		aa			n					ax			l	
				t	k			s	t		ow			n									
					t	eh		s	t		aa			ng								k	
					f	eh		s	t		ow			n									
dh	eh		ax	dh	t	eh	t	s	t	uw	ao		ah	n				k	eh	s		w	ay
dh	eh		ax	dh	t	eh		s	t	w	ao		ah	n				k	eh	s		w	ay
	eh			t		eh		s	t	w	ao		ax	n	t	hh			ae	t		w	ay

Figure 5.8: Phonetic alignment matrix based on articulatory-feature distance: (a) LTS phone sequences from HW recognition, (b) ensemble speech recognition phone sequence outputs, (c) HW LTS aligned segment, which is accurately time-bounded within the larger utterance.

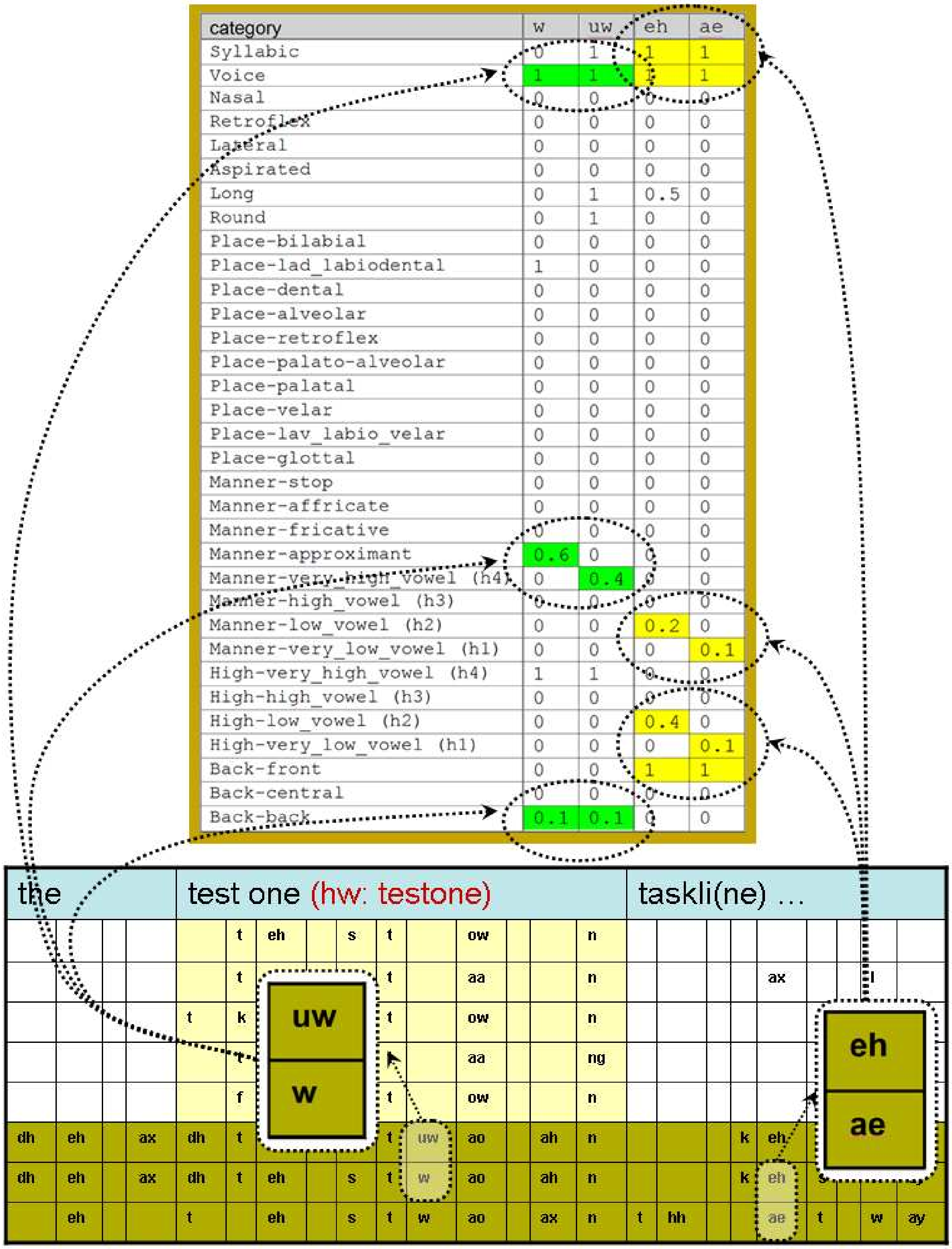
are stored is called a confusion matrix. The advantage of using a confusion matrix is that it is data driven and recognizer specific. The fact that it is recognizer specific is also a disadvantage, because if the vocabulary or language model of the recognizer changes then the confusion matrix needs to be recomputed. SHACER's goal as a dynamic learning system is to be constantly adding to the vocabulary and language model of both speech and handwriting recognizers. Therefore a recognizer specific confusion matrix within SHACER would have to be constantly recomputed. Kondrak's ALINE approach, based on static articulatory features that are not recognizer specific, out-performs simple Levenshtein edit distance [103] based on the spelling of phone symbols [97], and it also out-performs other articulatory-feature based alignment techniques that use only binary features [18, 138, 175] because of its assignment of saliency weights to the various categories of phonetic features [95, 96]. For example, the manner of articulation (e.g. stop, affricate, fricative, approximate, high/mid/low vowel) of two phones is generally more important in comparing them than considering their respective nasality or roundness, because nasality and roundness are features that only a few phones have. Therefore manner of articulation has a much greater saliency weight.

In Kondrak's algorithm some articulatory features are binary — roundness, voicing, syllabic, retroflex, lateral, aspirated, nasal. Some features are categorical — manner [*stop, affricate, fricative, approximate, high/mid/low vowel*], height [*high, mid-high, mid-low, low*], backness [*front, central, back*], and place [*bilabial, labiodental, dental, alveolar, retroflex, palato-alveolar, palatal, velar, uvular, pharyngeal, glottal*]. Vowels and consonants have different sets of active features. Each type in the sub-category set (in [*italics*]) of each of the four major features (manner, height, backness, place) has an assigned saliency weight based on empirical linguistic measurements (see Appendix A's Section A.2 and Table A.5). SHACER augments Kondrak's height feature to utilize four rather than three sub-categories, and in parallel with that adds a fourth vowel type to the manner feature (following Hosom [71]). So where Kondrak has *high*, *mid* and *low* manner features, SHACER has *very_high_vowel*, *high_vowel*, *low_vowel* and *very_low_vowel* manner features (see Section A.1).

In very recent work Kondrak has highlighted corpus-trained machine-learning approaches to determining phonetic distance, using either paired Hidden Markov Models or Dynamic Bayes Net (DBN) models. Both of these machine-learned distance models out-perform ALINE on cognate recognition tasks [97]. However, the drawback of a DBN machine-learning approach, like that of Filali and Bilmes [54], is that it requires a large training corpus. In the future, as larger multimodal handwriting/speech databases become available, such methods could be tried in SHACER and compared against the performance of its current salience-weighted articulatory-feature based distance measure.

Example of Phonetic Articulatory-Feature Based Alignment

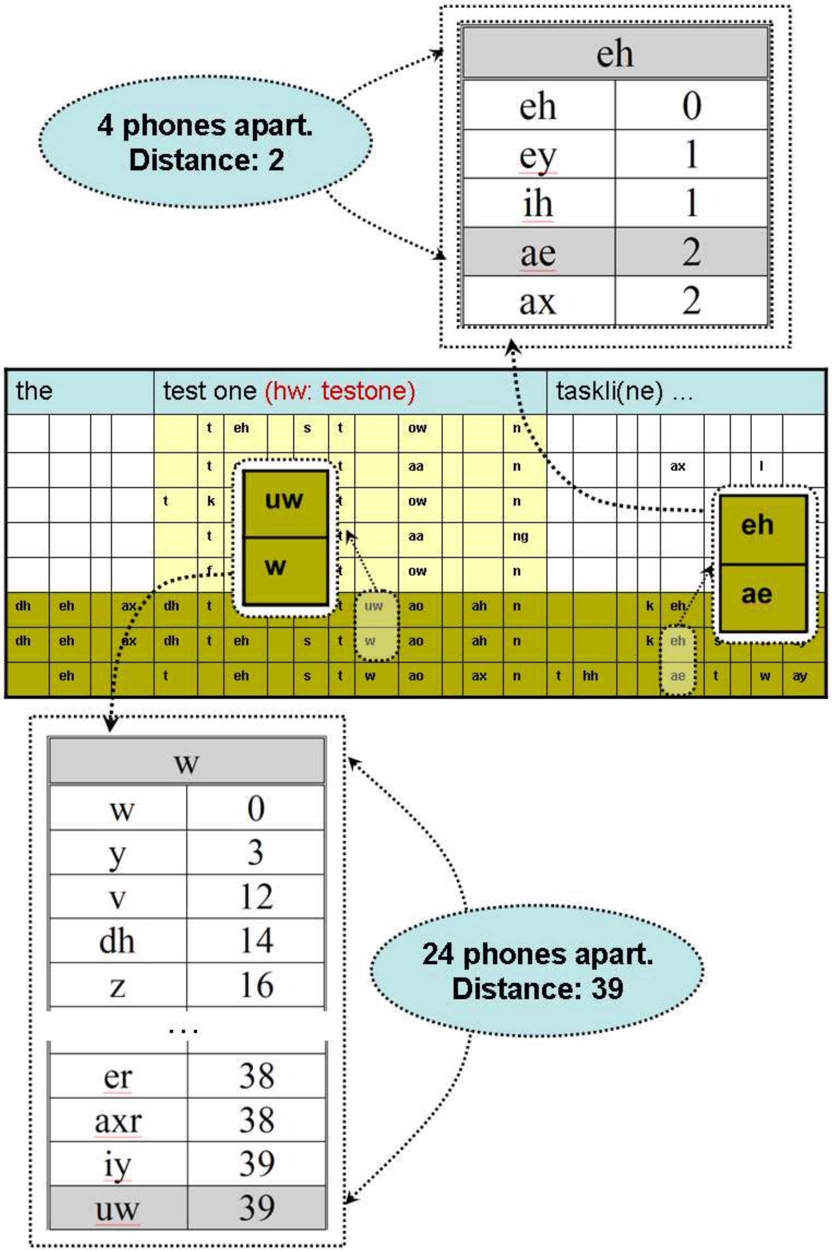
SHACER's phonetic articulatory-feature based aligner compares phone hypotheses by feature sets rather then by phone name, so instead of assigning the phone match between d and t an absolute score of 0 because they are not the same phone it can instead assign them a metric that takes into account the fact that they are identical in all articulatory features except voicing. Two further examples of how phonetic articulatory-feature-based alignment works are the eh/ae and w/uw alignments shown in Figures 5.9 and 5.10.



category	w	uw	eh	ae
Syllabic	0	1	1	1
Voice	1	1	1	1
Nasal	0	0	0	0
Retroflex	0	0	0	0
Lateral	0	0	0	0
Aspirated	0	0	0	0
Long	0	1	0.5	0
Round	0	1	0	0
Place-bilabial	0	0	0	0
Place-lad_labiodental	1	0	0	0
Place-dental	0	0	0	0
Place-alveolar	0	0	0	0
Place-retroflex	0	0	0	0
Place-palato-alveolar	0	0	0	0
Place-palatal	0	0	0	0
Place-velar	0	0	0	0
Place-lav_labio_velar	0	0	0	0
Place-glottal	0	0	0	0
Manner-stop	0	0	0	0
Manner-affricate	0	0	0	0
Manner-fricative	0	0	0	0
Manner-approximant	0.6	0	0	0
Manner-very_high_vowel (h4)	0	0.4	0	0
Manner-high_vowel (h3)	0	0	0	0
Manner-low_vowel (h2)	0	0	0.2	0
Manner-very_low_vowel (h1)	0	0	0	0.1
High-very_high_vowel (h4)	1	1	0	0
High-high_vowel (h3)	0	0	0	0
High-low_vowel (h2)	0	0	0.4	0
High-very_low_vowel (h1)	0	0	0	0.1
Back-front	0	0	1	1
Back-central	0	0	0	0
Back-back	0.1	0.1	0	0

the				test one (hw: testone)											taskli(ne) …								
					t	eh		s	t		ow			n									
					t				t		aa			n					ax			l	
				t	k				t		ow			n									
					t						aa			ng									
					f				t		ow			n									
dh	eh		ax	dh	t				t	uw	ao		ah	n				k	eh				
dh	eh		ax	dh	t	eh		s	t	w	ao		ah	n				k	eh	s			
	eh			t		eh		s	t	w	ao		ax	n	t	hh			ae	t		w	ay

Figure 5.9: The articulatory-feature table shows areas of similarity for *eh*/*ae* and *w*/*uw* example alignments. The *eh*/*ae* pair differ mostly in manner and height, while the *w*/*uw* pair differ in manner, place, long-ness and roundness.



eh	
eh	0
ey	1
ih	1
ae	2
ax	2

w	
w	0
y	3
v	12
dh	14
z	16
…	
er	38
axr	38
iy	39
uw	39

Figure 5.10: Distance scores from the underlying articulatory-feature table (shown in Figure 5.9) as well as salience values (Table A.1) to bias the weight put on each comparison feature, yield the phone distance look-up tables. Two partial phone-distance lookup table columns are shown here for *eh/ae* and *w/uw* example alignments. The *eh/ae* pair are much closer together phonetically than the *w/uw* pair.

The partial articulatory feature table in Figure 5.9 illustrates several areas of comparison for these two examples. See Appendix A for a full listing and explanation of the articulatory feature table used by SHACER. Both *eh* and *ae* are syllabic (by virtue of being vowels), and both are also voiced. As well, they are close in terms of their manner of articulation — *eh* being a low and *ae* being a very low vowel. They are close in their *backness* with both being *front* vowels. The scores from this articulatory feature table, as well as salience values to bias the weight put on each comparison, yield another table of phone distances like those exemplified in Figure 5.10. Fuller examples of phone distance tables are given in Section A.1. In Figure 5.10 the *AE* phone is 4 phones (out of 40) away from *EH* phone, while for the *w/uw* example the *UW* phone is 24 phones away from *W* phone with a correspondingly larger distance score.

5.3.4 The Mechanics Of Phonetic Alignment

The Effect of Seeding Alignment with Differing Sequences

```
* a-9 seed: ===================================
0.  a-9    #   f  ow   #   d   r  aw   n  ih  ng
1.  a-1    #   f  ao   r   b   r  aw   n  ih  ng
2.  a-2    #   f  ao   r   d   r  aw   n  ih  ng
3.  a-3    #  jh  ow   #   b   r  aw   n  ih  ng
4.  a-4    #   f  ao   r   b   r  aw   n  ih  ng
5.  a-5    #  jh  ow   #   d   r  aw   n  ih  ng
6.  a-6    #   f  ow   #   b   r  aw   n  ih  ng
7.  a-7    #   f  ao   r   b   r  aw   n  iy   z
8.  a-8    #  jh  ow   #   b   r  aw   n  ih  ng
9.  a-10  hh   #  ow   #   b   r  aw   n  ih  ng

* a-10 seed: ==================================
0.  a-10   #  hh  ow   #   b   r  aw   n  ih  ng
1.  a-1    f   #  ao   r   b   r  aw   n  ih  ng
2.  a-2    f   #  ao   r   d   r  aw   n  ih  ng
3.  a-3   jh   #  ow   #   b   r  aw   n  ih  ng
4.  a-4    f   #  ao   r   b   r  aw   n  ih  ng
5.  a-5   jh   #  ow   #   d   r  aw   n  ih  ng
6.  a-6    f   #  ow   #   b   r  aw   n  ih  ng
7.  a-7    f   #  ao   r   b   r  aw   n  iy   z
8.  a-8   jh   #  ow   #   b   r  aw   n  ih  ng
9.  a-9    f   #  ow   #   d   r  aw   n  ih  ng
```

Figure 5.11: Example alignments with different **seed** sequences. Upper alignment is seeded by *a-9* while lower alignment is seeded by *a-10*. Both alignments are of the same input set, but different seeding results in different alignments.

Currently we align all handwriting letter-to-sound (LTS) alternatives against each other first, before aligning them with the phone ensemble outputs, since we know that they refer to the same actual input. This yields a matrix of alignments like those shown in Figure 5.11's two examples. These two examples illustrate the point that depending upon which phone sequence is used as the first or seed phone sequence (e.g. *a-9* in the upper example and *a-10* in the lower example) different alignment matrices result. These alignment variations can effect the coherence of the resulting handwriting/speech alignment matrices — discussed in the next section.

Alignment Coherence

After aligning the handwriting (HW) alternatives as a block, each phone ensemble sequence in turn is then aligned against these HW blocks (as in the two examples in Figure 5.12). The resulting alignments can be characterized by a measure we have devised and named *coherence*, which compares phones within columns of the alignment matrix with a phone-to-phone articulatory-feature based distance metric (Appendix A).

The summed and averaged distances obtained from the *coherence* measuring procedure can be used to characterize the entire matrix. A matrix in which every row is exactly the same and all rows line up perfectly will be completely coherent. Thus it will have a *coherence* score of 1.0). Whereas a matrix in which no phones on any row align with any other phones will be almost completely in-coherent. Thus it will have a *coherence* score near 0.0. We are still very actively examining the best strategy for obtaining this *coherence* measure. For example, it often happens that the letter-to-sound transformations of the handwriting alternates are very coherent as a matrix by themselves, and this in turn could be considered as unduly biasing the overall coherence of the handwriting/speech matrices (like those Figure 5.12) by putting too little weight on the speech phone sequences in comparison to the handwriting alignment block.

We are experimenting with only comparing HW phones to speech phones and not to other HW phones during the coherence measurement, in an effort to arrive at a *coherence* measure of the HW/Speech matrix that is not unduly biased by the coherence of the HW-alternates block itself. However it is not yet clear if this is a better strategy or not. It may actually be useful to include the bias from the HW-alternates block coherence. More empirical work is needed to determine this. In Figure 5.12 *coherence* measures are given for the two alignments of HW/Speech matrices. Notice that the coherence measures differ for differing alignments. In this case suggesting that the first alignment, in which silence

```
   KEY: sp     = speech,
        hw-lts = handwriting letter-to-sound

                                                             (coherence)
        position   0   1   2   3   4   5   6   7   8   9  10   00.174
                 --- --- --- --- --- --- --- --- --- --- ---   ------
        |  0 ___   #   #   f  ow   #   d   r  aw   n  ih  ng
        |  1 ___   #   #   f  ao   r   b   r  aw   n  ih  ng
        |  2 ___   #   #   f  ao   r   d   r  aw   n  ih  ng
        |  3 ___   #   #  jh  ow   #   b   r  aw   n  ih  ng
        |  4 ___   #   #   f  ao   r   b   r  aw   n  ih  ng
hw-lts  |  5 ___   #   #  jh  ow   #   d   r  aw   n  ih  ng
phones  |  6 ___   #   #   f  ow   #   b   r  aw   n  ih  ng
        |  7 ___   #   #   f  ao   r   b   r  aw   n  iy   z
        |  8 ___   #   #  jh  ow   #   b   r  aw   n  ih  ng
        |  9 ___   #  hh   #  ow   #   b   r  aw   n  ih  ng

        | 10 ssb   #   #   w  ae   #   t   v   #   #   #   #
  sp    | 11 ssd   f   #   w  ae   t   p   s   #   #   #   #
phones  | 12 ssa   #   #   w  eh   #   t   s   #   #   #   #
        | 13 ssc   #   #   w  eh   #   t   s   #   #   #   #

          sframe   6   9  12  32  36  39  44  45  45  45  46

                                                            (coherence)
        position   0   1   2   3   4   5   6   7   8   9  10   00.154
                 --- --- --- --- --- --- --- --- --- --- ---   ------
        |  0 ___   #   #  hh  ow   #   b   r  aw   n  ih  ng
        |  1 ___   #   f   #  ao   r   b   r  aw   n  ih  ng
        |  2 ___   #   f   #  ao   r   d   r  aw   n  ih  ng
        |  3 ___   #  jh   #  ow   #   b   r  aw   n  ih  ng
        |  4 ___   #   f   #  ao   r   b   r  aw   n  ih  ng
hw-lts  |  5 ___   #  jh   #  ow   #   d   r  aw   n  ih  ng
phones  |  6 ___   #   f   #  ow   #   b   r  aw   n  ih  ng
        |  7 ___   #   f   #  ao   r   b   r  aw   n  iy   z
        |  8 ___   #  jh   #  ow   #   b   r  aw   n  ih  ng
        |  9 ___   #   f   #  ow   #   d   r  aw   n  ih  ng

        | 10 ssb   #   w   #  ae   t   v   #   #   #   #   #
  sp    | 11 ssd   f   w   #  ae   t   p   s   #   #   #   #
phones  | 12 ssa   #   w   #  eh   #   t   s   #   #   #   #
        | 13 ssc   #   w   #  eh   #   t   s   #   #   #   #

sframe   6   9  13  32  36  39  44  45  45  45  46
```

Figure 5.12: Example alignments of the block of letter-to-sound handwriting transformations (rows preceded by ___) from Figure 5.11 with the phone ensemble alternatives (rows preceded by *ss_* labels). The varying alignments resulting from different seed sequences result in different coherence measures (upper right of each alignment block).

— i.e., # — is mostly at the beginning of the rows, is somewhat more coherent (score = *0.174* on a scale from *0.0* to *1.0*) than the second (score = *0.154*), in which silence occurs more often internally, away from the start column of the matrix. This is the kind of distinction we want coherence to capture and represent.

The top row of the two displays in Figure 5.12 each is an index of sequence positions. Each position identifies a column in the matrix, and has an associated start frame, which is shown in the *sframe* row at the bottom of each matrix. Notice that for the more coherent matrix (the upper one) the first phone of *Joe Browning* (i.e., *jh*) tends to start around frame *12* at position *2*, whereas for the less coherent matrix (the lower one) it tends to start around frame *9* at position *1*. To estimate the start frame for each position we average the start/end frame information from each phone ensemble output. For the syllabic ensemble sequences we use interpolation to suggest syllable-internal phonetic boundaries. However, as these examples show the timing estimates (in terms of start/stop frames) are only as good as the alignments themselves.

```
         0  1  2  3  4  5  6  7  8  9 10 11 12 13 14  00.677 (coherence)
        -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  ------
  0 ___  #  f  # ao  r  b  #  r aw  n  # iy  #  z  #
  1 ___  #  f  # ao  r  b  #  r aw  n  # ih ng  #  #
  2 ___  #  f  # ao  r  d  #  r aw  n  # ih ng  #  #
  3 ___  # jh  # ow  #  b  #  r aw  n  # ih ng  #  #
  4 ___  #  f  # ao  r  b  #  r aw  n  # ih ng  #  #
  5 ___  # jh  # ow  #  d  #  r aw  n  # ih ng  #  #
  6 ___  #  f  # ow  #  b  #  r aw  n  # ih ng  #  #
  7 ___  # jh  # ow  #  b  #  r aw  n  # ih ng  #  #
  8 ___  #  f  # ow  #  d  #  r aw  n  # ih ng  #  #
  9 ___ hh  #  # ow  #  b  #  r aw  n  # ih ng  #  #
 10 ssb  # sh uw ow  #  p  #  r aw  m dh iy ng  d  m
 11 ssd  # sh  y uw  l  b  p  r aw  m dh iy ng  d  m
 12 ssa  # sh  # uw  w  b  #  r aw  n  # ih ng  #  #
 13 ssc  # sh  # uw  w  w  # er aw  n  # iy ng  #  #
```

Figure 5.13: Coherent example alignment of handwritten *Joe Browning* with spoken, “Joe Browning.”

The alignments in Figure 5.12 are not very coherent, and indeed they should not be, because the handwriting for *Joe Browning* is being aligned against the phone-ensemble sequences for the utterance, ”Let’s ...”. The alignment in Figure 5.13 of the handwriting for *Joe Browning* with the phone-ensemble sequences for the speech, “Joe Browning,” is a much more coherent matrix (score = *0.677*). Setting a threshold on the acceptable value of the *coherence* metric defines one of SHACER’s primary means for deciding whether the aligned handwriting and speech are actually redundant. Low coherence alignments are disregarded. High coherence alignments trigger further processing under the assumption that redundancy may have occurred.

5.3.5 Alignment Procedure Modifications for SHACER

SHACER's alignment routine is based largely on the definitions and algorithm given in Kondrak [96, 95] for his ALINE technique. However, for SHACER we have modified Kondrak's algorithm in several ways. We have added a capability to split the alignment of diphthongs in a way that makes sense for the alignment task that we are performing. Each diphthong has two sub-phones of which it is composed. A diphthong's distance from another phone can be measured based either on (1) the diphthong's full set of articulatory features or on (2) the set of articulatory features belonging to either one of its sub-phone members.

Diphthong Expansion

During alignment a diphthong can be expanded to cover the space of its compound members (or of other phones that are very close to those compound members) as they occur in other rows of input. For example, the diphthong, *ey*, shown in the *Partial Alignment Matrix* in Figure 5.14 (line *7*), has been expanded in this way. This expansion is shown in close-up in the *Select lines from Partial Alignment Matrix* section of Figure 5.14, in which the rows 4,7, and 12 of the *Partial Alignment Matrix* have been selected, extracted and grouped. It can be seen that the *ey* diphthong in row *7* has been expanded into component pieces that represent its compound phones (*first_phone* = *eh*, and *second_phone* = *iy*). The expanded pieces are labeled in ways that identify them as part of the *ey* diphthong while also denoting their roles in the expansion — i.e., the first_phone/second_phone sequence *_ey ey* representing the diphthong's compound parts.

The first part of the *ey* expansion (e.g., *_ey*) represents the phone *eh* and in the *Select lines from Partial Alignment Matrix* section of Figure 5.14 is aligned below the phone *ae*. In Figure 5.14's *Partial Section from Phone-Distance Table* section (a larger example of the phone distance table is given in Section A.6) it can be seen that the phone *ae* is the closest phone to *eh*. This closeness drives the algorithmic expansion of the diphthong, *ey*, into its component parts, because the *ae*/*eh* closeness scores better than the closeness of *ey* as a diphthong to any other phone in lines of *1-6* of Figure 5.14's *Partial Alignment Matrix*. Below the second of *ey*'s compound expansion phones (symbolized by *ey* immediately after *_ey*) is *iy* which is an exact phonetic match to the diphthong's second compound member.

```
Partial Alignment Matrix
0  ___   #   #   #   r   #  ow
1  ___   f  ao   #   r   #   #
2  ___   z  ih   #   r   #  ow
3  ___   #  ae   #   z   #   #
4  ___   #  ae   #   d   #   #
5  ___   #  ih   #   z   #   #
6  ___   #   #   #   d   #  iy
7  ___   # _ey  ey   #   #   #
8  ___   #  ah   #   #   #   #
9  ssb   y  iy  iy   d   t  uw
10 ssd   y   #  ey   t   y  uw
11 ssa   n   #  iy   d   t  uw
12 ssc   n   #  iy   d   t  uw

Select lines from Partial Alignment Matrix
4  ___   #  ae   #   d   #   #
7  ___   # _ey  ey   #   #   #
12 ssc   n   #  iy   d   t  uw

Partial Section from Phone-Distance Table
|  28 ae         |
|----------------|
|  28 ae     0   |
|  27 eh     2   |
|  26 ey     3   |
|  25 ih     3   |
```

Figure 5.14: An example of diphthong expansion — row 7 *_ey ey*.

It is important in our coherence metric to have such expansions, because non-phones (e.g. silence or noise — the # symbol) occurring in any matrix match-segment internal columns degrade the coherence. Expanding the diphthongs allows our coherence metric to better take into account the acoustic properties of diphthongs. Kondrak's algorithm, designed for studying cognate change during language evolution, does not explicitly support diphthong expansion in this way. Our approach is akin to the use of expansions in other dynamic-programming-based matching routines described in the literature [152].

Handling Length Mis-matches in Phone Alignments

For the *Joe Browning* alignment example (Figure 5.13) the spoken utterance, "Joe Browning," and the LTS phones from the handwriting input, *Joe Browning*, were close to the same length. However, this length correspondence is not always the case, as shown in the example matrix in Figure 5.15 of which the first 12 frames have been truncated to allow it to fit on this page. In this example *Joe Browning* is being aligned with the utterance, "This is our timeline for *Joe Browning*." So the *Joe Browning* reference array of letter-to-sound generated phone sequences from the handwriting is much shorter than the phone ensemble sequences, covering only positions 20-32 (leaving positions 0-20 unmatched as is appropriate in this case).

```
       13 14 15 16 17 18 19 20 21 22 23 24 25 26  27 28 29 30 31 32 33  00.616
       -- -- -- -- -- -- -- -- -- -- -- -- -- --- -- -- -- -- -- -- --  ------
0 ssd   # ao ay  n  f  r  d jh  y uw  l  p  r   # aw  #  m  # ih ng  d
1 ssa   l oy ih  n  f er  d jh  # uw  l  b  r _aw aw  #  n  # ih ng  #
2 ssb   # oy  #  n  f er  d jh  y uw  l  p  r  ae ah  #  n  # ih ng  #
3 ssc   l oy  #  n  f er  d jh  # uw  l  b  r   # aw  #  n  # ih ng  #
4 ___   #  #  #  #  #  #  # jh  # ow  #  b  r   # ah  #  n  # ih ng  #
5 ___   #  #  #  #  #  #  # jh  # ow  #  b  r   # aw  t  #  # ih ng  #
6 ___   #  #  #  #  #  #  # jh  # ow  #  k  r   # ah  #  n  # ih ng  #
7 ___   #  #  #  #  #  #  # jh  # ow  #  k  r   # aw  t  #  # ih ng  #
8 ___   #  #  #  #  #  #  # jh  # ow  #  b  r   # aw  #  n  r ih ng  #
```

Figure 5.15: Example of length mis-matched alignment of letter-to-sound phoneme sequences for *Joe Browning* compared to ensemble phone output for utterance, "(This is our time) line for Joe Browning." The spoken portion in parentheses has been truncated so the example will fit this page.

In Figure 5.15 the speech phones in rows *0-3* serve as the *reference* strings for the alignment, while the handwriting LTS phones in rows *4-8* serve as the *hypothesis* strings. This mis-match in reference and hypothesis string lengths also arises during the handwriting/speech-transcript letter-sequence matching task, as shown in Figure 5.16's *Buy Computer* example. This letter matching task is part of the comparison of the handwriting to the speech transcript, and it is also part of the comparison of handwriting to lattice word sequence extractions that will be discussed in the next section. To keep the phones or letters from a handwriting instance relatively close together and not too far spread out the cost of insertion/deletion moves is increased, as shown in Section **??** at lines *a-d* of the *dynamic_programming_match* pseudo-code.

```
a. buYcomputer________
b. buTcomputerANDOTHER

a. buSYcomputer________
b. buT_computerANDOTHER
```

Figure 5.16: Example alignment of handwriting letter sequence (a) compared to letter concatenations of the LVCSR transcript (b). This letter alignment is also performed in evaluating the multiparser's request for phonary alignment. Upper-case letters below an underscore are *insertions*. Upper-case letters above an underscore are *deletions*. Upper-case letters below another upper-case letter are *substitutions*.

Global Versus Local Optimal Dynamic Programming Alignments

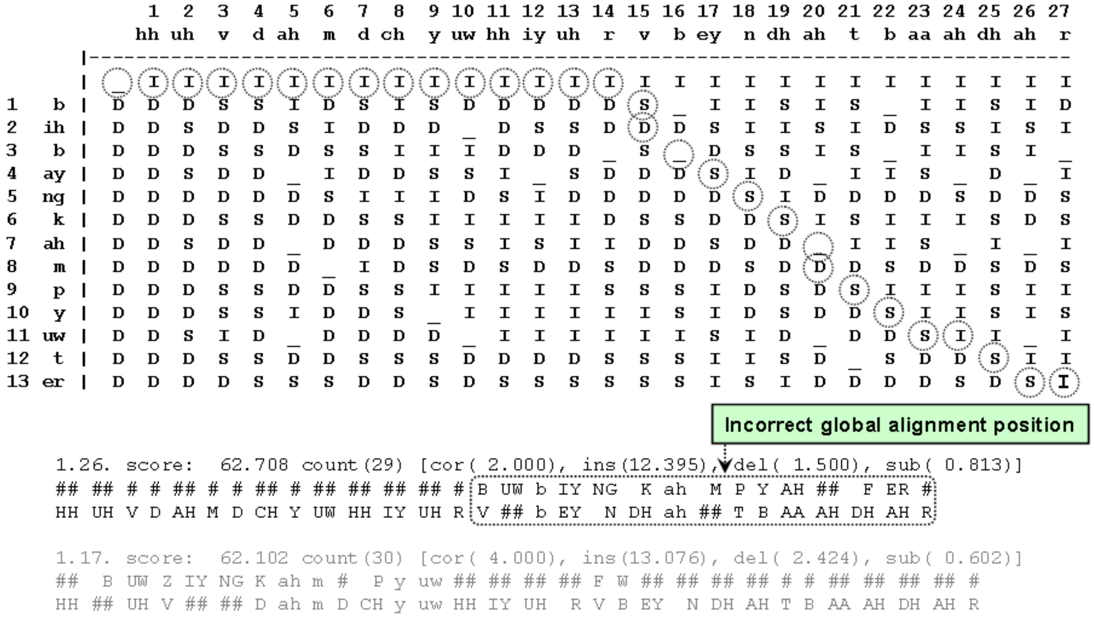
Aside from keeping the handwriting's constituent letters or phones close together during the dynamic programming matching routines, it is also necessary to check the finished dynamic programming matrix for the best local matches. For example, Figures 5.17, 5.18 and 5.19 below show the alignment matrix for handwritten *Buy Computer* and a phone recognizer output for the associated utterance, "...buy computer and other."

First, a portion of the full matrix, with best-move and best-score information is shown in Figure 5.17. Each matrix cell is labeled by a tuple that shows the best move to that cell along with score of that best move. The move possibilities are I=INSertion, D=DELetion, S=SUBstitution, __=___correct. These best-moves and best-scores represent all of the information that is typically recorded in the memory structures of the dynamic programming pass. It is possible to implement memory structures that would keep a fuller listing of the matrix, accounting for not only best-move/best-score information but for all possible scores of all four move possibilities at each matrix cell, which is similar to what occurs during the Viterbi search that is central to speech recognition. However, we have found that back-tracing along the edges of the finished matrix (e.g. right-to-left along the bottom row and bottom-to-top along the right-most column) can yield the optimal local path for length mis-matched alignments, which as shown in Figures 5.18 and 5.19 is not always the same as the global best path.

The first of the full matrices for this example, shown in Figure 5.18, shows only the best moves resulting from the dynamic programming match algorithm. The best path through this DP matrix is circled, and the path's score, statistics and alignment are shown highlighted below the figure. This is the best global path found by the algorithm. However, it is possible to look for alternative local solutions by iteratively replacing each move on the final column (moving from bottom to top) by a Deletion. Then, as each replacement is made, the best path is re-computed. Likewise each move along the bottom row (from right to left) can be replaced by an Insertion move, and the best local path re-computed. When we do these iterations for this matrix we do find a better scoring local path as we add Insertions from right-to-left along the bottom row. This best local path is shown in the Figure 5.19 with its path statistics highlighted below the figure. Lower path scores are better, so the local path score of *62.102* is better than the global score of *62.708*. These statistics show that the local path's better score is primarily due to more correct matches. In this case the alignment of the handwritten *Buy Computer* against the spoken, "Buy computer and other," is moved from below the " *... and other*" portion of

```
Phone ensemble phones (horizontal axis):
hh uh v d ah m d ch y uw hh iy uh r v b ey n dh ah t b aa ah dh ah r
-- -- - -
Rough association of speech with ensemble phones:
|  buy  |       ... computer ...       |  and  |   ... other ...    |


Handwriting LTS phone alignment (vertical axis):

3  2  1  > | top edge of aligned recognizer alternates
-- -- -- - | ---------------------------------------
   b  >  > | b
   ih >  > | ih
b  z  b  > | b
ah iy ay > | ay
ng >  >  > | ng
k  k  k  > | k
ah ah ah > | ah
m  m  m  > | m
p  p  p  > | p
y  y  y  > | y
uw uw uw > | uw
t  t  t  > | t
er er er > | er

                      (1 hh)      (2 uh)      (3 v)       (4 d)       ...
          (_ 000.000) (I 000.000) (I 000.000) (I 000.000) (I 000.000) ...
(1    b)  (D 000.000) (D 000.000) (D 000.000) (S 023.000) (S 029.000) ...
(2   ih)  (D 000.000) (D 000.000) (S 011.333) (D 008.000) (D 014.000) ...
(3    b)  (D 000.000) (D 000.000) (D 000.000) (S 033.667) (S 033.667) ...
(4   ay)  (D 000.000) (D 000.000) (S 008.775) (D 018.667) (D 018.667) ...
(5   ng)  (D 000.000) (D 000.000) (D 000.000) (D 003.667) (D 003.667) ...
(6    k)  (D 000.000) (D 000.000) (D 000.000) (S 001.000) (S 018.667) ...
(7   ah)  (D 000.000) (D 000.000) (S 009.750) (D 000.000) (D 003.667) ...
(8    m)  (D 000.000) (D 000.000) (D 000.000) (D 000.000) (D 000.000) ...
(9    p)  (D 000.000) (D 000.000) (D 000.000) (S 013.000) (S 019.000) ...
(10   y)  (D 000.000) (D 000.000) (D 000.000) (S 015.000) (S 022.000) ...
(11  uw)  (D 000.000) (D 000.000) (S 015.550) (I 000.550) (D 007.000) ...
(12   t)  (D 000.000) (D 000.000) (D 000.550) (S 027.950) (S 024.150) ...
(13  er)  (D 000.000) (D 000.000) (D 000.000) (D 012.950) (S 016.150) ...
```

Figure 5.17: A portion of the full dynamic programming finished matrix for the alignment of the handwritten *Buy Computer* and phone ensemble output for the spoken utterance, "…buy computer and other." The phone-level hypothesis is shown at the top. The first four phones of that hypothesis, which appear in the truncated horizontal axis of the matrix, are underlined. A rough association of the spoken utterance words to the phones is given. The vertical axis of the alignment matrix is composed of the top edge of the alignment of the handwriting recognizer's LTS alternates (labeled *3,2,1*), which is listed vertically with ">" signs denoting the top (i.e. rightmost) edge. This edge is what is shown as the vertical axis of the alignment matrix. Key: D = Deletion, I = Insertion, S = Substitution, _ = Correct.



```
                 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27
                hh  uh   v   d  ah   m   d  ch   y  uw  hh  iy  uh   r   v   b  ey   n  dh  ah   t   b  aa  ah  dh  ah   r
          |-----------------------------------------------------------------------------------------------------------------
          |   _   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
 1      b |   D   D   D   S   S   I   D   S   I   S   D   D   D   D   D   S   _   I   I   S   I   S   _   I   I   S   I   D
 2     ih |   D   D   S   D   D   S   I   D   D   D   _   D   S   S   D   D   D   S   I   I   S   I   D   S   S   I   S   I
 3      b |   D   D   D   S   S   D   S   S   I   I   I   D   D   D   _   S   _   D   S   S   I   S   _   I   I   S   I   _
 4     ay |   D   D   S   D   D   _   I   D   D   S   S   I   _   S   D   D   D   S   I   D   _   I   I   S   _   D   _   I
 5     ng |   D   D   D   D   D   D   S   I   I   I   D   S   I   D   D   D   D   D   S   I   D   D   D   D   S   D   D   S
 6      k |   D   D   D   S   S   D   D   S   S   I   I   I   I   I   D   S   S   D   D   S   I   S   I   I   I   S   D   S
 7     ah |   D   D   S   D   D   _   D   D   D   S   S   I   S   I   I   D   D   S   D   D   _   I   I   S   _   I   _   I
 8      m |   D   D   D   D   D   D   _   I   D   S   D   S   D   D   S   D   D   D   S   D   D   D   S   D   D   S   D   S
 9      p |   D   D   D   S   S   D   D   S   S   I   I   I   I   I   S   S   S   I   D   S   D   S   I   I   I   S   I   I
10      y |   D   D   D   S   S   I   D   D   S   _   I   I   I   I   I   I   S   I   D   S   D   D   S   I   I   S   I   S
11     uw |   D   D   S   I   D   _   D   D   D   D   _   I   I   I   I   I   I   S   I   D   _   D   D   S   I   I   _   I
12      t |   D   D   D   S   S   D   D   S   S   S   D   D   D   D   S   S   S   I   I   S   D   _   S   D   D   S   I   I
13     er |   D   D   D   D   S   S   S   D   D   S   D   S   S   S   S   S   S   I   S   I   D   D   D   D   S   D   S   I
```

Incorrect global alignment position

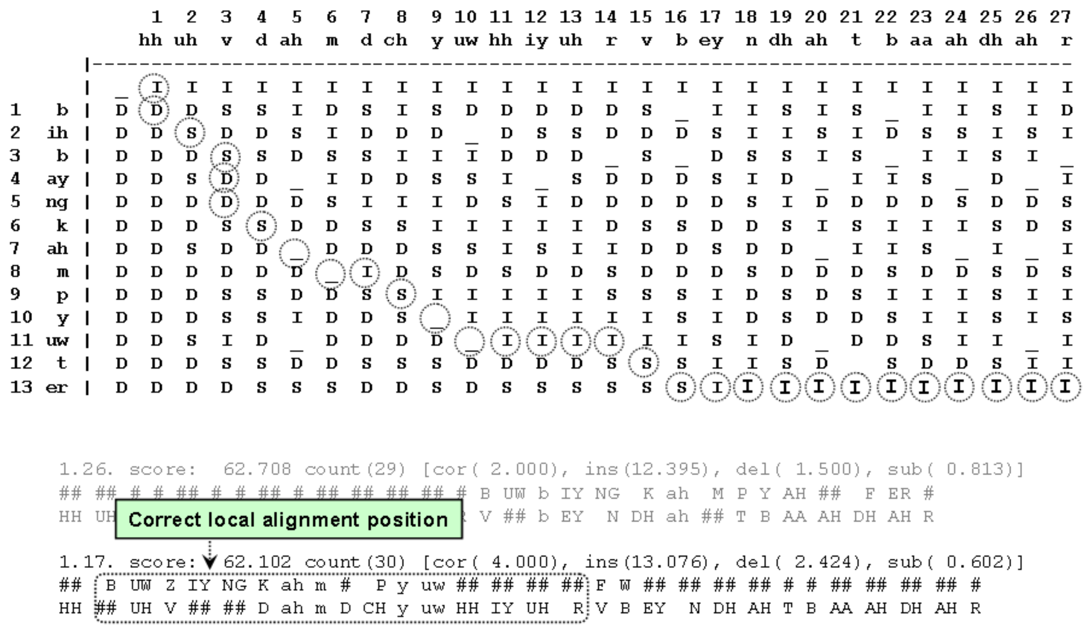
```
1.26. score:  62.708 count(29) [cor( 2.000), ins(12.395), del( 1.500), sub( 0.813)]
## ## # # ## # # ## # ## ## ## ## # B UW b IY NG  K ah  M P Y AH ##  F ER #
HH UH V D AH M D CH Y UW HH IY UH R V ## b EY  N DH ah ## T B AA AH DH AH R

1.17. score:  62.102 count(30) [cor( 4.000), ins(13.076), del( 2.424), sub( 0.602)]
##  B UW Z IY NG K ah m #  P y uw ## ## ## ## F W ## ## ## ## # # ## ## ## ## #
HH ## UH V ## ## D ah m D CH y uw HH IY UH  R V B EY  N DH AH T B AA AH DH AH R
```

Figure 5.18: Example of global best path for length mis-matched alignment of letter-to-sound phoneme sequences for *Buy Computer* compared to ensemble phone output for utterance, “a computer and other.” Circles trace the path of best states backwards from the lower right corner.

the speech to below the “*Buy computer ...*” portion of the speech, where it should be.

Alignment Frame-Level Timing

As alignments are being processed the approximate frame-time of each column position within the aligned matrices is determined by an averaging mechanism. The alignment matrices are then cropped by removing phonetic outliers. These outliers are shown in the *pruned* area of Figure 5.20. They arise from (a) errors in handwriting recognition, (b) errors in letter-to-sound transformation, or (c) errors that are combinations of these two processes. The outliers are pruned when they are more than a factor of standard deviation away from the main body of the phone alignment matrix. Figure 5.20 offers an illustration of this process. The phones *r ah* in row *5* and *aa* in row *8* (in position columns *19* and *20*) are all pruned away. Their distance from the main body of the alignment, which is marked off with vertical dividers between positions *24* and *28*, is about three full positions away — from position *20* to position *24*. This exceeds the standard deviation in distances between phones that occur within the main alignment body. In this case that standard



```
            1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
           hh uh  v  d ah  m  d ch  y uw hh iy uh  r  v  b ey  n dh ah  t  b aa ah dh ah  r
        |---------------------------------------------------------------------------------
        |  _  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I  I
1    b  |  D  D  D  S  S  I  D  S  I  S  D  D  D  D  D  S  _  I  I  S  I  S  _  I  I  S  I  D
2   ih  |  D  D  S  D  D  S  I  D  D  D  _  D  S  S  D  D  D  S  I  I  S  I  D  S  S  I  S  I
3    b  |  D  D  D  S  S  D  S  S  I  I  I  D  D  D  _  S  _  D  S  S  I  S  _  I  I  S  I  _
4   ay  |  D  D  S  D  D  _  I  D  D  S  S  I  _  S  D  D  D  S  I  D  _  I  I  S  _  D  _  I
5   ng  |  D  D  D  D  D  D  S  I  I  I  D  S  I  D  D  D  D  D  S  I  D  D  D  D  S  D  D  S
6    k  |  D  D  D  S  S  D  D  S  S  I  I  I  I  I  D  S  S  D  D  S  I  S  I  I  I  S  D  S
7   ah  |  D  D  S  D  D  _  D  D  D  S  S  I  S  I  I  D  D  S  D  D  _  I  I  S  _  I  _  I
8    m  |  D  D  D  D  D  D  _  I  D  S  D  S  D  D  S  D  D  D  S  D  D  D  S  D  D  S  D  S
9    p  |  D  D  D  S  S  D  D  S  S  I  I  I  I  I  S  S  S  I  D  S  D  S  I  I  I  S  I  I
10   y  |  D  D  D  S  S  I  D  D  S  _  I  I  I  I  I  I  S  I  D  S  D  D  S  I  I  S  I  S
11  uw  |  D  D  S  I  D  _  D  D  D  D  _  I  I  I  I  I  I  S  I  D  _  D  D  S  I  I  _  I
12   t  |  D  D  D  S  S  D  D  S  S  S  D  D  D  D  S  S  S  I  I  S  D     S  D  D  S  I  I
13  er  |  D  D  D  D  S  S  S  D  D  S  D  S  S  S  S  S  S  I  I  I  I  I  I  I  I  I  I  I
```

```
1.26. score:  62.708 count(29) [cor( 2.000), ins(12.395), del( 1.500), sub( 0.813)]
## ## # # ## # # ## # ## ## ## ## # B UW b IY NG  K ah  M P Y AH ##  F ER #
HH UH                                V ## b EY  N DH ah ## T B AA AH DH AH R

1.17. score: 62.102 count(30) [cor( 4.000), ins(13.076), del( 2.424), sub( 0.602)]
## B UW Z IY NG K ah m #  P y uw ## ## ## ## F W ## ## ## ## # # ## ## ## ## #
HH ## UH V ## ## D ah m D CH y uw HH IY UH  R V B EY  N DH AH T B AA AH DH AH R
```

Correct local alignment position

Figure 5.19: Example of local best path for length mis-matched alignment of letter-to-sound phoneme sequences for *Buy Computer* compared to ensemble phone output for utterance, “a computer and other.” Circles trace the path of best states backwards from the lower right corner, with replacement insertion moves along the bottom row.

deviation is between *1-2* positions.

Figures 5.21 and 5.22 illustrate an alignment matrix with its accompanying phone-level timing information. Each line of phonetic frame-level start times shown in Figure 5.22 corresponds to the phone hypothesis for the accompanying ensemble phone recognizer in Figures 5.21. As mentioned earlier, SHACER currently has no mechanism for extracting syllable internal temporal, so the *ssa* and *ssc* lines in Figure 5.22, which correspond to phone hypotheses generated by syllable-based phone recognizers, have no syllable-internal timing information about phonetic boundaries. The two phonotactically constrained phone recognizers (*ssb* and *ssd* in Figure 5.22) are basically identical to word-level speech recognizers, except that words are replaced by individual phones. Thus they do provide phone-level start/end times. To extract frame-level temporal information SHACER averages across all four phone ensemble timing matrices. Currently temporal phone boundaries within syllables are interpolated. In our future work we intend to adopt the use of speech recognizers that not only better recognize phonetic structure (e.g. syllabic-based phone recognizers) but also support the extraction of the phone-level temporal information that SHACER

```
          19  20  21  22  23 | 24  25  26  27  28 | 29  30
         --- --- --- --- --- |--- --- --- --- --- |--- ---
 0 ssb    r  uh   r   f   # |  r  ih   #   #  hh | er   w
 1 ssa    r  uh   r   f  er |  r  ih   l  jh   # | er   w
 2 ssc    r  uh   r   f  er |  r  ih   l  jh   # | er   w
 3 ssd    y   #  er   f   # |  r  ih   #   v  uh |  r   w
 4 ___    #   #   #   #  er |  #  ay   #   v   # |  #   #
 5 ___    r  ah   #   #   m |  #   #   #   #   # |  #   #
 6 ___    #   #   #   #   # | er  ih   #   v  ah |  #   #
 7 ___    #   #   #   #   # | er  ay   #   v   # |  #   #
 8 ___    #  aa   #   #   # |  r  uw   #   d  ah |  #   #
 9 ___    #   #   #   #   # | er  ay   #   v   # |  #   #
10 ___    #   #   #   #   # | er  ay   #   v   # |  #   z
11 ___    #   #   #   #   # | er  ih   #   v  ah |  #   #
12 ___    #   #   #   #   # | er  ay   #   v   # | er   #
13 ___    #   #   #   #   # | er  uw   #   b  ah |  #   #

          .. (pruned).. |   | .... "arrive" .... |
```

Figure 5.20: An alignment illustration for the handwritten word, *arrive*, which shows how phonetic outliers are pruned. The pruning occurs for (*r ah* in row *5* and *aa* in row *8*. These sequences occupy position columns *19* and *20*.

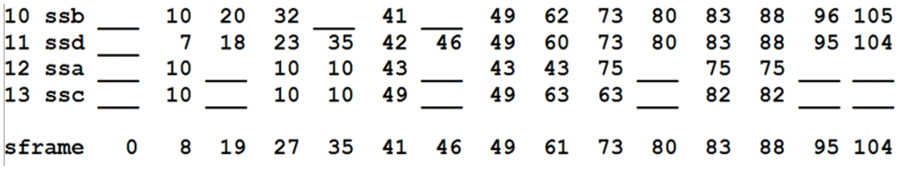
needs for the next step in its dynamic learning of new words, which is *refinement* of the new word's pronunciation.

```
    pos   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
    --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
 7  ___   #  jh   #  ow   #   b   #   r  aw   n   #  ih   #   #  ng
 8  ___   #   f   #  ow   #   d   #   r  aw   n   #  ih   #   #  ng
 9  ___  hh   #   #  ow   #   b   #   r  aw   n   #  ih   #   #  ng
10  ssb   #  sh  uw  ow   #   p   #   r  aw   m  dh  iy  ng   d   m
11  ssd   #  sh   y  uw   l   b   p   r  aw   m  dh  iy  ng   d   m
12  ssa   #  sh   #  uw   w   b   #   r  aw   n   #  ih  ng   #   #
13  ssc   #  sh   #  uw   w   w   #  er  aw   n   #  iy  ng   #   #
```

Figure 5.21: A portion of the phone alignment matrix (for handwritten and spoken, *Joe Browning*). Rows 7-9 are the 7th-9th handwriting letter-to-sound hypotheses. Rows 10-13 are ensemble phone recognizer outputs.

5.4 REFINEMENT OF HANDWRITING/SPEECH SEGMENT PRONUNCIATION

After phonetically aligning redundant handwriting and speech, as described above, the next step is second-pass phone recognition. One of the phone ensemble recognizers acts as a *master* phone recognizer. Its first function is to produce a phonetic sequence hypothesis, which is routed to the multiparser where it participates in the alignment process used to



```
10 ssb ___   10  20  32 ___  41 ___  49  62  73  80  83  88  96 105
11 ssd ___    7  18  23  35  42  46  49  60  73  80  83  88  95 104
12 ssa ___   10 ___  10  10  43 ___  43  43  75 ___  75  75 ___ ___
13 ssc ___   10 ___  10  10  49 ___  49  63  63 ___  82  82 ___ ___

sframe   0    8  19  27  35  41  46  49  61  73  80  83  88  95 104
```

Figure 5.22: The parallel start-frame matrix for the ensemble phone recognizers: **ssb/ssd** use phone sub-word units, while **ssa/ssc** use syllables. The bottom row is the output start-frame vector for the hypothesized phone positions, averaged over the phone sub-word unit recognizers. Note that the syllable sub-word unit recognizers have no unit-internal phone segmentations.

discover redundancies. Its second function is to cache the MEL-cepstrum features from the first pass recognition and then re-use them for a fast second pass recognition to refine the pronunciation of any discovered redundancy. Second pass recognition using cached features is constrained by a phone sequence model built from the alignment matrices. The alignment matrix also designates the temporal segment of a spoken utterance that corresponds to the handwriting. This is why finding the optimal *local* alignment path is so important. For example, the optimal *global* path in Figure 5.18 defines the temporal segment of speech for second pass recognition to be that segment in which the speaker said, "...and other," while the optimal *local* path in Figure 5.19 defines the temporal segment of speech for second pass recognition to be that segment in which the speaker said, "...buy computer." Second pass recognition on the *global* path segment (for "...and other") would not find a good phonetic pronunciation for the handwritten *Buy Computer*, while second pass recognition on the *local* path segment (for "...buy computer") will find a refined pronunciation. Only the alignment-designated temporal segment is searched during second pass recognition. This segment's temporal boundaries can also be used to identify word sequences from the transcribing recognizer's lattice, which may provide further matching evidence in comparison with the handwriting input, as described next.

5.4.1 Lattice Term Sequence Extraction

An instance in which *Joe Browning* is redundantly handwritten and spoken yields the alignment shown in Figure 5.23. The letter-to-sound phone sequences from the handwriting alternates list are in rows *0-9*, and the phone ensemble sequence outputs are in rows *10-13*.

The bottom row in the diagram shown in Figure 5.23 is labeled *sframe* and it lists the

```
           0   1   2   3   4   5   6   7   8   9  10  11  12  13  14    00.446
         --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---    ------
  0 ___    #   f   #  ow   #   d   #   r  aw   n   #  ih  ng   #   #
  1 ___    #   f   #  ao   r   b   #   r  aw   n   #  ih  ng   #   #
  2 ___    #   f   #  ao   r   d   #   r  aw   n   #  ih  ng   #   #
  3 ___    #  jh   #  ow   #   b   #   r  aw   n   #  ih  ng   #   #
  4 ___    #   f   #  ao   r   b   #   r  aw   n   #  ih  ng   #   #
  5 ___    #  jh   #  ow   #   d   #   r  aw   n   #  ih  ng   #   #
  6 ___    #   f   #  ow   #   b   #   r  aw   n   #  ih  ng   #   #
  7 ___    #   f   #  ao   r   b   #   r  aw   n   #  iy   z   #   #
  8 ___    #  jh   #  ow   #   b   #   r  aw   n   #  ih  ng   #   #
  9 ___   hh   #   #  ow   #   b   #   r  aw   n   #  ih  ng   #   #
 10 ssb    #  sh  uw  ow   #   p   #   r  aw   m  dh  iy  ng   d   m
 11 ssd    #  sh   y  uw   l   b   p   r  aw   m  dh  iy  ng   d   m
 12 ssa    #  sh   #  uw   w   b   #   r  aw   n   #  ih  ng   #   #
 13 ssc    #  sh   #  uw   w   w   #  er  aw   n   #  iy  ng   #   #
 sframe    0   8  19  27  35  41  46  49  61  73  80  83  88  95 104
```

Figure 5.23: An example alignment matrix for handwritten and spoken *Joe Browning.*

start-frame approximations for each alignment column, which are a result of averaging timing information from each of the phone ensemble inputs. In this example since the handwriting and speech are the same, the alignment segment is the entire alignment, but this is not always the case, as in the *Buy Computer* example discussed above.

```
  0.    1  91  2         JOE BROWN  || jh ow + b r aw n
  1.    1  91  3    JOE BROWN MEAN  || jh ow + b r aw n + m iy n
  2.    1  91  3    JOE BROWN RING  || jh ow + b r aw n + r ih ng
  3.    1  91  2        SHOW BROWN  || sh ow + b r aw n
  4.    1  91  3   JOE BROWN MEANS  || jh ow + b r aw n + m iy n z
  5.    1  91  3    JOE BROWN DING  || jh ow + b r aw n + d ih ng
  6.    1  91  3      JOE BROWN E.  || jh ow + b r aw n + iy
...
 81.    1  91  3     JOE BRAND TEA  || jh ow + b r ae n d + t iy
 82.    1  91  3     SHOW BRAND A.  || sh ow + b r ae n d + ey
 83.    1  91  3    SHOW BRAND AIM  || sh ow + b r ae n d + ey m
 84.    1  91  3     SHOW BRAND T.  || sh ow + b r ae n d + t iy
 85.    1  91  3    SHOW BRAND TEA  || sh ow + b r ae n d + t iy
```

Figure 5.24: Extracted LVCSR lattice sequences based on the temporal boundaries of the alignment matrix in Figure 5.23. Column 1 numbers the extractions, column 2 is the start frame, column 3 is the endframe, and column 4 is the number of words in the extraction. The right-most column is the canonical pronunciation of each extracted sequence.

The alignment segment's average start/stop times (e.g. start=8/stop=88 from the *sframe* line in Figure 5.23) are used to query the LVCSR lattice for the utterance being processed and extract all possible word sequences over that alignment segment's time boundaries. SHACER uses an iterative back-off on the segment boundaries, starting with the tightest (which is a slight constriction of the alignment boundaries) and progressively widening the boundaries until a threshold limit of word sequences is reached. For example, for the alignment matrix of *Joe Browning* as shown in Figure 5.23, the top seven lattice extractions and the last five lattice extractions out the eighty-five possible word sequences extracted from the lattice are shown in Figure 5.24. Notice that because the word, *Browning*, was not in the dictionary the correct word is not actually present in the lattice; however, there is a lot of phonetic information in the extracted word sequences that are present in this lattice segment.

```
ID sf ef CombProb Cohere LAlign PAlign MatchLS Handwriting ||        Lattice
-- -- -- -------- ------ ------ ------ ------- -----------    ---------------
0   1 91    0.770  0.790  0.727  0.755   0.944 JoeBrowning ||  JOE_BROWN_RING
1   1 91    0.762  0.789  0.727  0.748   0.944 JoeBrowning ||  JOE_BROWN_DING
2   1 91    0.738  0.791  0.636  0.758   0.885 JoeBrowning || JOE_BROWN_THING
3   1 91    0.724  0.727  0.636  0.639   0.885 JoeBrowning || JOE_BROWN_RINGS
4   1 91    0.718  0.729  0.545  0.630   0.859 JoeDrowning ||  JOE_BROWN_RING
5   1 91    0.714  0.727  0.545  0.617   0.859 FoeBrowning ||  JOE_BROWN_RING
6   1 91    0.710  0.727  0.545  0.623   0.859 JoeDrowning ||  JOE_BROWN_DING
7   1 91    0.710  0.726  0.545  0.638   0.859 JoeBrowning ||  JOE_BROWN_DINK
8   1 91    0.707  0.728  0.545  0.644   0.859 JoeBrowning ||  JOE_BROWN_LINK
9   1 91    0.706  0.725  0.545  0.610   0.859 FoeBrowning ||  JOE_BROWN_DING
```

Figure 5.25: Re-ranked LVCSR lattice extractions. The extraction sequence is shown in the right-most column labeled *Lattice*.

To take advantage of the phonetic information in these extracted lattice sequences, they are compared to the handwriting sequences. Based on that comparison they are then re-ranked on the following scales:

1. The coherence of their phonetic alignments (see the *Cohere* for *Coherence* column in Figure 5.25).

2. Their letter and phone alignment scores (see the *LAlign* for *Letter Align* and *PAlign* for *Phone Align* columns in Figure 5.25).

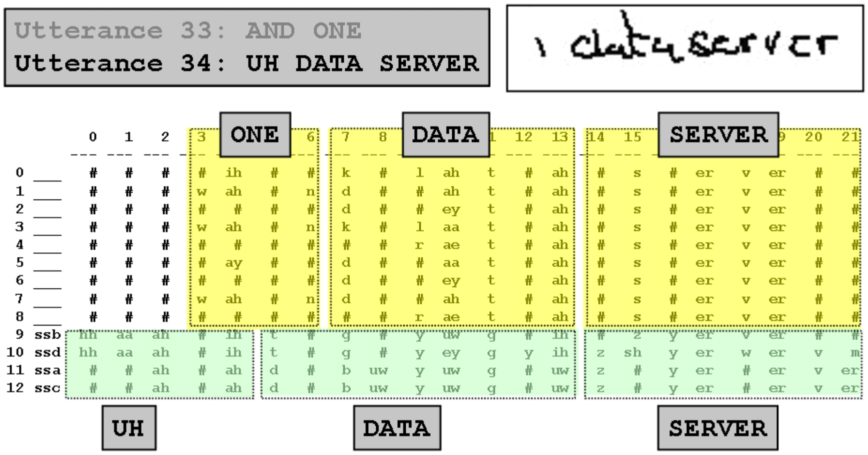
3. Their spelling closeness, which is measured as the percentage of matching letters between the closest handwriting/lattice-words combination (see the *MatchLS* for *Match Letter Score* column in Figure 5.25).

These various scores are combined into a single probability that is computed as a weighted average (see the *CombProb* for *Combined Probability* column in Figure 5.25), by which the lattice sequences are ranked. A threshold on this combined probability determines which of these lattice sequences are themselves grouped and phonetically aligned.

Then, as an aligned group, they are aligned against the ensemble speech outputs. This creates further alignment matrices, which can be mined for further phone sequence information to constrain second pass recognition.

5.4.2 Positional Phone-Bigram Modeling

To consolidate the phone sequence information available from both the speech and handwriting input streams (Figure 5.21) we have designed and implemented a technique that we call Positional Phone-Bigram Modeling. Figures 5.26 and 5.27 together give an example of positional phone bigram modeling. For the alignment matrix shown in Figure 5.26 the user spoke, "And one . . . uh data server." This speech was segmented into two short utterances. As the user spoke these utterances he also wrote the term *1 dataserver*.



Utterance 33: AND ONE
Utterance 34: UH DATA SERVER

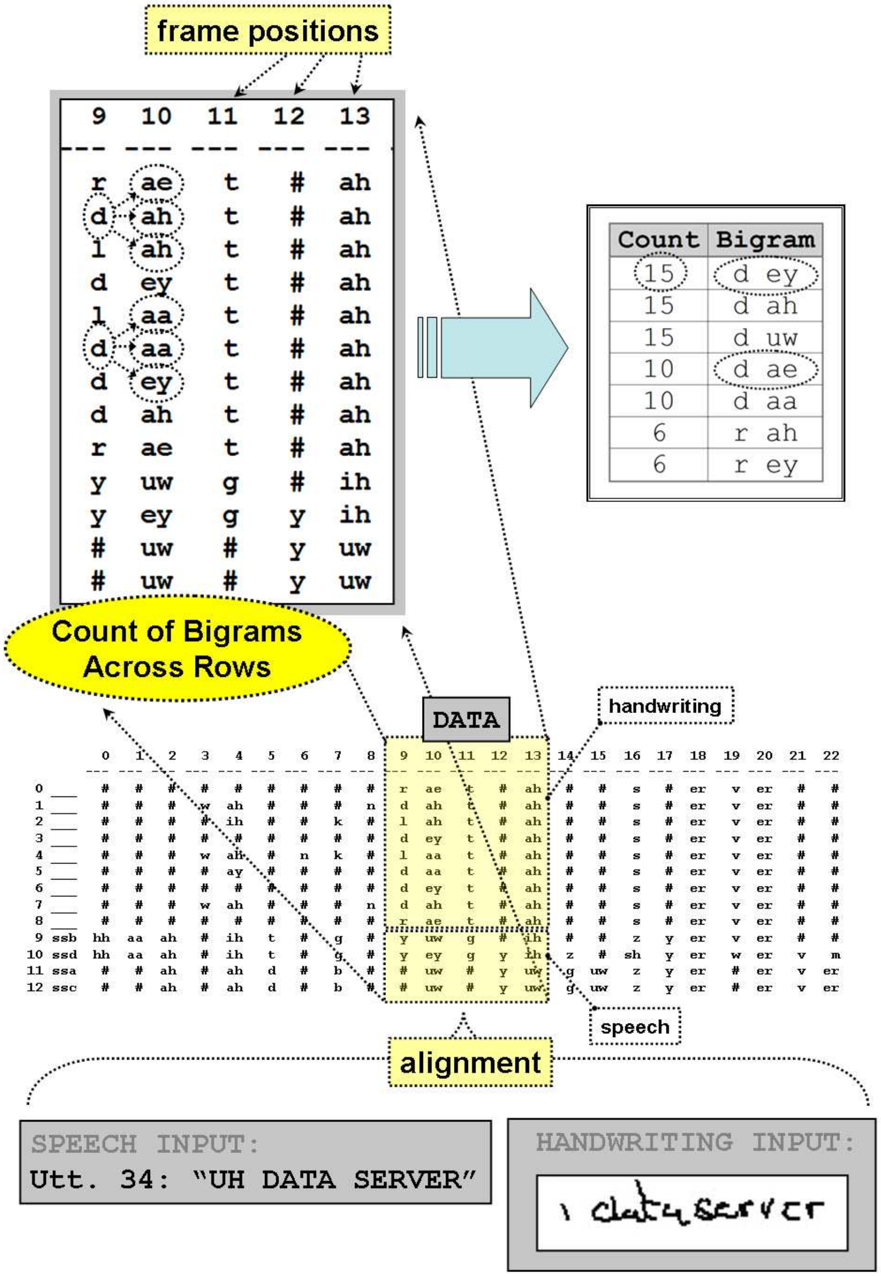
		0	1	2	3			6	7	8			1	12	13	14	15				9	20	21
0	___	#	#	#	#	ih	#	#	k	#	l	ah	t	#	ah	#	s	#	er	v	er	#	#
1	___	#	#	#	w	ah	#	n	d	#	#	ah	t	#	ah	#	s	#	er	v	er	#	#
2	___	#	#	#	#	#	#	#	d	#	#	ey	t	#	ah	#	s	#	er	v	er	#	#
3	___	#	#	#	w	ah	#	n	k	#	l	aa	t	#	ah	#	s	#	er	v	er	#	#
4	___	#	#	#	#	#	#	#	#	#	r	ae	t	#	ah	#	s	#	er	v	er	#	#
5	___	#	#	#	#	ay	#	#	d	#	#	aa	t	#	ah	#	s	#	er	v	er	#	#
6	___	#	#	#	#	#	#	#	d	#	#	ey	t	#	ah	#	s	#	er	v	er	#	#
7	___	#	#	#	w	ah	#	n	d	#	#	ah	t	#	ah	#	s	#	er	v	er	#	#
8	___	#	#	#	#	#	#	#	#	#	r	ae	t	#	ah	#	s	#	er	v	er	#	#
9	ssb	hh	aa	ah	#	ih	t	#	g	#	y	uw	g	#	ih	#	z	y	er	v	er	#	#
10	ssd	hh	aa	ah	#	ih	t	#	g	#	y	ey	g	y	ih	z	sh	y	er	w	er	v	m
11	ssa	#	#	ah	#	ah	d	#	b	uw	y	uw	g	#	uw	z	#	y	er	#	er	v	er
12	ssc	#	#	ah	#	ah	d	#	b	uw	y	uw	g	#	uw	z	#	y	er	#	er	v	er

Figure 5.26: Phonetic alignment of speech and handwriting information. The user spoke two utterances: (1) "And one" . . . (then he paused and said) (2) "Uh data server." As he spoke he wrote *1 dataserver*.

SHACER does not yet handle instances where the handwriting corresponds to portions of more than one utterance, as is the case for the handwriting in Figure 5.26. However, our exploration of the delivery timing of multimodal redundancies across handwriting and speech (Section 2.2.3) does indicate some guidelines for how we can go about this in our future work. For example, typically handwriting overlaps temporally with all utterances to which it should be aligned. If a single instance of handwriting overlaps both (a) an earlier-starting spoken utterance and (b) a later-finishing spoken utterance, then that suggests that those two utterances could be treated as one for the purpose of alignment with handwriting, to discover any redundancy. In Figure 5.26 only the second spoken utterance is aligned with the handwriting, so there is a partial mismatch between the aligned speech and handwriting. The handwriting section of the alignment is the upper row blocks in Figure 5.26. It represents the phonetic letter-to-sound-generated sequence from the handwritten term, "one data server." The speech phone sequences are shown in the lower row blocks of Figure 5.26, and represent the utterance, "uh data server."

In Figure 5.27 another alignment matrix for this same combination is shown. Recall that many such matrices are constructed and used during processing, because their alignments vary depending on the first or seed phone sequence. Phone bigram counts are collected from all alignment matrices for a given attempt at handwriting and speech integration. In Figure 5.27 we highlight the area that corresponds to the word *data*, and see how information from that matrix can help in modeling the likelihood of how the word *data* was pronounced (e.g. either *d ae t ah* or *d ey t ah*). Our dynamic positional phone bigram counts bigrams across rows, as shown in the enlarged box on the upper-left side of Figure 5.27. This results in bigram counts like those shown in Figure 5.27's right-side *count/bigram* table, where in comparing between *d ey t ah* and *d ae t ah* the former is more likely based on phone sequence information extracted from this matrix.

Equations 1 and 2 in Figure 5.28 show how the positional bigram counts are computed (the interpolated normalization method is not shown). Equation 1 in Figure 5.28 states that for each phone (p), which is a member of the phone set (P), and for each frame position (j) from 0 to the number of columns (c) in the alignment matrix (e.g. Figure 5.21), the count of bigram *(p1,p2)* at frame/column position j (i.e., *ct(j,p1,p2)*) is summed over all occurrences of *p1* in the i'th row (i from 0 to the number of rows, r, in the matrix) and *p2* in k'th row of the l'th column such that neither *p1* nor *p2* is silence (s = silence). Note that l ranges from *j+1* to *cns*, where the *ns* subscript means the nearest column with a non-silence phone. The notation *p(i,j)* denotes the phone at the i'th row and j'th column of the matrix. The handling of silence in bigrams at the beginning and end of each matrix row is not shown, and otherwise silence is not allowed be part of a bigram. Thus in Equation 1 in Figure 5.28 if a phone is silence (s) then it is not considered, and the bigram in which it is participating does not add to the bigram count. Positional information, j, keeps track of the start frame of the bigram's first phone (which is computed from the average start frames of the phone sub-unit recognizers, as shown in Figure 5.22). This is used to constrain the bigram not to be used beyond a threshold of distance from its start position. The intuitive motivation is that bigrams from the end of an alignment segment may not be appropriate to use near the beginning of an alignment segment.



```
  9   10   11   12   13
 ---  ---  ---  ---  ---
  r   ae    t    #   ah
  d   ah    t    #   ah
  l   ah    t    #   ah
  d   ey    t    #   ah
  l   aa    t    #   ah
  d   aa    t    #   ah
  d   ey    t    #   ah
  d   ah    t    #   ah
  r   ae    t    #   ah
  y   uw    g    #   ih
  y   ey    g    y   ih
  #   uw    #    y   uw
  #   uw    #    y   uw
```

Count	Bigram
15	d ey
15	d ah
15	d uw
10	d ae
10	d aa
6	r ah
6	r ey

```
          0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22
         --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
 0 ___    #   #   #   #   #   #   #   #   #   r  ae   t   #  ah   #   #   s   #  er   v  er   #   #
 1 ___    #   #   #   w  ah   #   #   #   n   d  ah   t   #  ah   #   #   s   #  er   v  er   #   #
 2 ___    #   #   #   #  ih   #   #   k   #   l  ah   t   #  ah   #   #   s   #  er   v  er   #   #
 3 ___    #   #   #   #   #   #   #   #   #   d  ey   t   #  ah   #   #   s   #  er   v  er   #   #
 4 ___    #   #   #   w  ah   #   n   k   #   l  aa   t   #  ah   #   #   s   #  er   v  er   #   #
 5 ___    #   #   #   #  ay   #   #   #   #   d  aa   t   #  ah   #   #   s   #  er   v  er   #   #
 6 ___    #   #   #   #   #   #   #   #   #   d  ey   t   #  ah   #   #   s   #  er   v  er   #   #
 7 ___    #   #   #   w  ah   #   #   #   n   d  ah   t   #  ah   #   #   s   #  er   v  er   #   #
 8 ___    #   #   #   #   #   #   #   #   #   r  ae   t   #  ah   #   #   s   #  er   v  er   #   #
 9 ssb   hh  aa  ah   #  ih   t   #   g   #   y  uw   g   #  ih   #   #   z   y  er   v  er   #   #
10 ssd   hh  aa  ah   #  ih   t   #   g   #   y  ey   g   y  ih   z   #  sh   y  er   w  er   v   m
11 ssa    #   #  ah   #  ah   d   #   b   #   #  uw   #   y  uw   g  uw   z   y  er   #  er   v  er
12 ssc    #   #  ah   #  ah   d   #   b   #   #  uw   #   y  uw   g  uw   z   y  er   #  er   v  er
```

SPEECH INPUT:
Utt. 34: "UH DATA SERVER"

HANDWRITING INPUT:

Figure 5.27: An illustration of how Positional Phone-Bigram Modeling works. As shown in the box on the upper-left, phone sequences are counted across rows in the alignment matrix. These counts are then used to weight second-pass phone recognition over saved speech features. For example, given the phone-bigrams available in this alignment matrix for the word *data*, the *d ey t ah* pronunciation appears more likely than the *d ae t ah* pronunciation. This is shown in *count/bigram* table in the upper-right.

$$\left\{\begin{aligned} &\forall p \in P, p_1 \neq s, p_2 \neq s, \forall j \middle| 0 \leq j \leq c : \\ &\quad ct(j, p_1, p_2) = \sum_{i=0}^{r} \sum_{l=j+1}^{c_{ns}} \sum_{k=0}^{r} bg(j, p(i,j), p(k,l))) \end{aligned}\right\} \quad (1)$$

$$bg(j, p(i,j), p(k,l)) = \begin{cases} 0 & p_1 \neq p(i,j) \\ 0 & p_2 \neq p(k,l) \\ 1 & else \end{cases} \qquad (2)$$

Figure 5.28: Positional phone bigram equations.

Positional Phone Bigram Model Example

Approximate frame-times across all alignment matrices support the construction of a positional bigram model of phone sequences as described in Section 5.4.2 above. For the handwriting, *Buy Computer*, and partially redundant speech, "A computer and other . . . ," various alignment matrices like that shown in Figure 5.29 contribute phone sequence information to a positional phone bigram model for constraining second pass phone-level recognition.

```
         0   1   2   3   4   5   6   7   8   9  10  11  12  13
       --- --- --- --- --- --- --- --- --- --- --- --- --- ---
 0 ___   #   #   #   b  ay   #   k  ah   m   #   p   y  uw   #
 1 ___   #   #   #   b  ay   #   k  ah   m   #   p   y  uw   #
 2 ___   #   b  ih   z  iy   #   k  ah   m   #   p   y  uw   #
 3 ___   #   #   #   b  ay   #   k  ah   m   #   p   y  uw   #
 4 ___   #   b  eh   r  iy   #   k  ah   m   #   p   y  uw   #
 5 ___   #   #   #   b  ay   #   k  ah   m   #   p   #  ah   #
 6 ___   #   b  uw   #  iy   #   k  ah   m   #   p   y  uw   #
 7 ___   #   #   #   b  ay   #   k  ah   m   #   p   #  ah   #
 8 ___   #   #   #   b  ah  ng   k  ah   m   #   p   y  uw   #
 9 ___   #   #   #   b  ay   #   k  ah   m   #   p   y  uw   #
10 ssb  hh   #  uh   v   #   #   d  ah   m   d  ch   y  uw  hh
11 ssd   #   #  ao   v   #   #   g  uw   m   b  ch   y  uw  hh
12 ssa   #   b  ah   p  ih  ng   k   #   #   #   f   y  uw  uw
13 ssc   #   f  ah   v   #   #   k  ah   m   k   f   y  uw  uw
------ --- --- --- --- --- --- --- --- --- --- --- --- --- ---
sframe  21  23  25  30  32  34  37  42  46  54  58  68  75  80
```

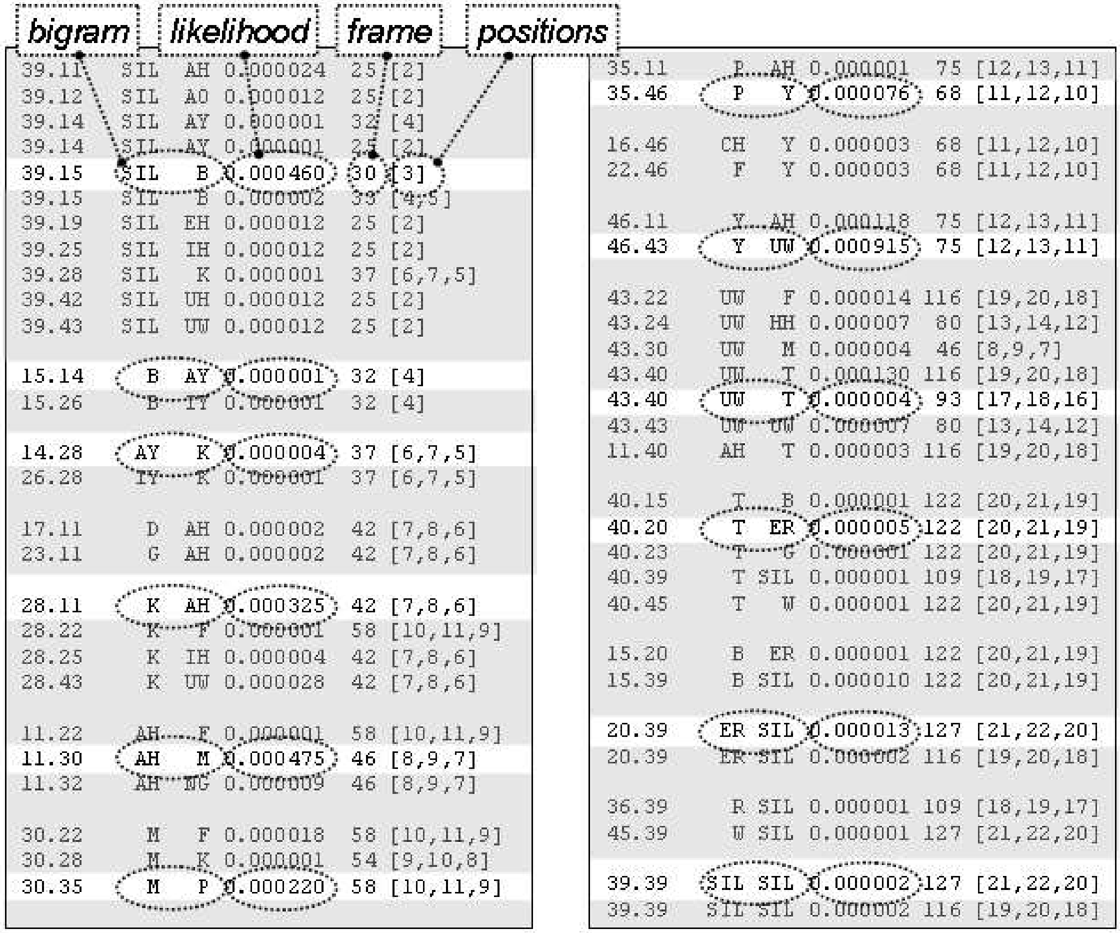
Figure 5.29: A section from one alignment matrix for the handwriting, *Buy Computer*, and the partially redundant utterance, "A Computer and other . . . ." Handwritten information is in rows *0-9*, while ensemble phone recognition information is in rows *10-13*. Phone sequences collected from this partial alignment matrix and others like it for this pair of inputs are the basis for the top-scoring bigrams shown in Figure 5.30.

For the alignment request, which is partially shown above in Figure 5.29, actual bigram likelihoods are shown in Figure 5.30. For space reasons, only the upper range of normalized bigram likelihoods are shown in Figure 5.30. This range include bigrams with likelihoods greater than 0.1e-5. From left-to-right each row in Figure 5.30 shows for each bigram (1) the pair of phone IDs, (2) the pair phone names (Figure 5.30, *bigram* column), (3) the bigram likelihood (Figure 5.30, *likelihood* column), (4) the average frame distance from the start of the aligned segment (Figure 5.30, *frame* column), and finally (5) the list of positions at which the bigram occurs (Figure 5.30, *positions* column). Figure 5.29's *sframe* row depicts how positions relate to the alignment matrix).

The correct bigrams in Figure 5.29 are highlighted, and ordered in sequence from top-to-bottom. For this example the highlighted bigrams have the highest likelihoods compared to other possible bigrams at their respective positions. The actual bigram model used for second-pass constraint is an empirically weighted interpolation of (a) these illustrated bigram statistics along with (b) unigram statistics for all phones appearing in all contributing alignment matrices and (c) a small amount of probability mass for all other non-appearing phones.

5.4.3 Second Pass Phone Recognition

As mentioned earlier, the effect of using the positional phone bigram model during second pass phone recognition is analogous to that of using a word-level language model to constrain the acoustic choices made by an LVCSR speech recognizer. In both cases the sequence model biases the scores of known and frequently encountered sequences to be higher than the scores of those sequences that are unknown or have not occurred frequently in the data on which the model was trained. The positional phone bigram model holds combined information from (1) the phone ensemble, (2) the handwriting letter-to-sound transformations, (3) the lattice extractions and possibly also from (4) exact transcript matches. These constraints on phone sequencing then interact with the actual acoustics of a spoken utterance. Section 3.2.3 introduced an example of a handwritten and redundantly spoken proper name, *Fred Green.* The speech recognition transcript for this input was *Fred's Green*, with an inserted possessive — because the sequence *Fred Green* was not in the language model training data. Second pass recognition, using the positional phone bigram-model, yielded the correct pronunciation (e.g. *F R EH D G R IY N*). In the refined, second-pass pronunciation there was no incorrectly inserted possessive *'s* between *Fred* and *Green.* This is the value of second-pass pronunciation refinement.



```
39.11   SIL  AH 0.000024  25 [2]
39.12   SIL  AO 0.000012  25 [2]
39.14   SIL  AY 0.000001  32 [4]
39.14   SIL  AY 0.000001  25 [2]
39.15   SIL   B 0.000460  30 [3]
39.15   SIL   B 0.000002  33 [4,5]
39.19   SIL  EH 0.000012  25 [2]
39.25   SIL  IH 0.000012  25 [2]
39.28   SIL   K 0.000001  37 [6,7,5]
39.42   SIL  UH 0.000012  25 [2]
39.43   SIL  UW 0.000012  25 [2]

15.14     B  AY 0.000001  32 [4]
15.26     B  IY 0.000001  32 [4]

14.28    AY   K 0.000004  37 [6,7,5]
26.28    IY   K 0.000001  37 [6,7,5]

17.11     D  AH 0.000002  42 [7,8,6]
23.11     G  AH 0.000002  42 [7,8,6]

28.11     K  AH 0.000325  42 [7,8,6]
28.22     K   F 0.000001  58 [10,11,9]
28.25     K  IH 0.000004  42 [7,8,6]
28.43     K  UW 0.000028  42 [7,8,6]

11.22    AH   F 0.000001  58 [10,11,9]
11.30    AH   M 0.000475  46 [8,9,7]
11.32    AH  NG 0.000009  46 [8,9,7]

30.22     M   F 0.000018  58 [10,11,9]
30.28     M   K 0.000001  54 [9,10,8]
30.35     M   P 0.000220  58 [10,11,9]
```

```
35.11     P  AH 0.000001  75 [12,13,11]
35.46     P   Y 0.000076  68 [11,12,10]

16.46    CH   Y 0.000003  68 [11,12,10]
22.46     F   Y 0.000003  68 [11,12,10]

46.11     Y  AH 0.000118  75 [12,13,11]
46.43     Y  UW 0.000915  75 [12,13,11]

43.22    UW   F 0.000014 116 [19,20,18]
43.24    UW  HH 0.000007  80 [13,14,12]
43.30    UW   M 0.000004  46 [8,9,7]
43.40    UW   T 0.000130 116 [19,20,18]
43.40    UW   T 0.000004  93 [17,18,16]
43.43    UW  UW 0.000007  80 [13,14,12]
11.40    AH   T 0.000003 116 [19,20,18]

40.15     T   B 0.000001 122 [20,21,19]
40.20     T  ER 0.000005 122 [20,21,19]
40.23     T   G 0.000001 122 [20,21,19]
40.39     T SIL 0.000001 109 [18,19,17]
40.45     T   W 0.000001 122 [20,21,19]

15.20     B  ER 0.000001 122 [20,21,19]
15.39     B SIL 0.000010 122 [20,21,19]

20.39    ER SIL 0.000013 127 [21,22,20]
20.39    ER SIL 0.000002 116 [19,20,18]

36.39     R SIL 0.000001 109 [18,19,17]
45.39     W SIL 0.000001 127 [21,22,20]

39.39   SIL SIL 0.000002 127 [21,22,20]
39.39   SIL SIL 0.000002 116 [19,20,18]
```

Figure 5.30: Selected portions of the complete bigram statistics listing for alignment of the handwriting, *Buy Computer*, and speech, "A computer and other . . . " A partial alignment matrix for these inputs is shown in Figure 5.29. Note that in this example the correct bigrams (highlighted and circled) receive the most weight.

5.5 SUMMARY

This chapter has described the *alignment* and *refinement* steps in SHACER's detection and learning of OOV terms. These are the first two of SHACER's three main processing steps. The third main processing step is *integration*, which uses the refined pronunciations described in this chapter as a distance metric for choosing the spelling and pronunciation of a redundantly presented OOV term. *Integration* will be fully described in the next chapter.

We compared SHACER's approach to learning new words to the problem of retrieving relevant documents in the Spoken Document Retrieval task. SDR tasks use either a vector space model (VSM) approach or a dynamic programming (DP) matching approach for deciding which documents are relevant. For small document databases DP is significantly better, although slower. Comparing handwriting against a small window of previous spoken utterances, as SHACER does, is equivalent to a small database SDR task. Therefore SHACER's use of a DP approach is reasonable. Further experimentation is called for in the future to determine if SHACER can benefit from a hybrid VSM/DP approach.

We reviewed recent literature showing that machine-learning techniques for discovering phonetic distances do out-perform static articulatory-feature tables like that which SHACER uses. However, since these machine-learning approaches are recognizer specific they would have to be re-computed whenever new words were added to the vocabulary and language model of the system. We argue that since SHACER's purpose is exactly that — to be constantly discovering and adding new words to the system — a static articulatory-feature distance definition is appropriate. In the future the necessity and computational cost of constantly updating machine-learned phone distances will have to be further evaluated within SHACER.

This chapter described SHACER's use of articulatory-feature based alignment, and detailed the differences between SHACER's implementation and the original algorithm on which is was patterned. Those differences were (a) diphthong expansion and (b) optimal DP local path discovery.

Finally, SHACER's use of the articulatory-feature based alignment matrices to support phone-sequence modeling was described. SHACER's phone-sequence modeling approach is called Positional Phone-Bigram Modeling. It plays a critical role in constraining second pass speech recognition over cached speech features, which is the core of SHACER's *refinement* processing step.

Chapter 6

SHACER: Integration

As discussed in chapter 3, SHACER has three main functionalities: (1) *alignment*, (2) *refinement*, and (3) *integration*. The previous chapter discussed *alignment* and *refinement*. This chapter discusses *integration*. *Integration* uses the refined pronunciation from step 2 as an *integration decision metric* against which to compare other inputs, and decide on the best combination of spelling, pronunciation and semantics. This integration step differentiates SHACER from SDR approaches, which do not attempt to learn new words in this way.

6.1 INPUT SOURCES

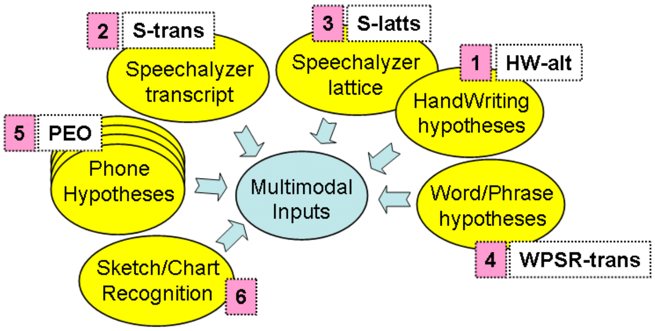


Figure 6.1: A diagram of the various multimodal inputs that contribute to SHACER's integration of redundant handwriting and speech information. (KEY: WPSR-trans = Word/Phrase-Spotting Recognizer transcript, S-trans = speech-transcript, S-latts = speech-lattices, HW-alt = handwriting alternates list, PEO = Phone Ensemble Output).

There are five information sources across which comparisons are made: (1) handwriting letter-strings and their phone sequences, (2) LVSCR word-level transcripts, (3) LVSCR word-level lattices, (4) word/phrase-spotter recognitions, (5) ensemble phone-level recognitions. These information sources are pictured graphically in Figure 6.1, which also shows a sixth information source — sketch/chart recognition. The sixth input provides context in which combinations of the other five input sources are interpreted. The diagram of input sources (Figure 6.1) will be used throughout this chapter to provide a common perspective on the examples discussed. Each numbered input is labeled with its abbreviation, and these abbreviations are the same as those used in Figure 6.2.

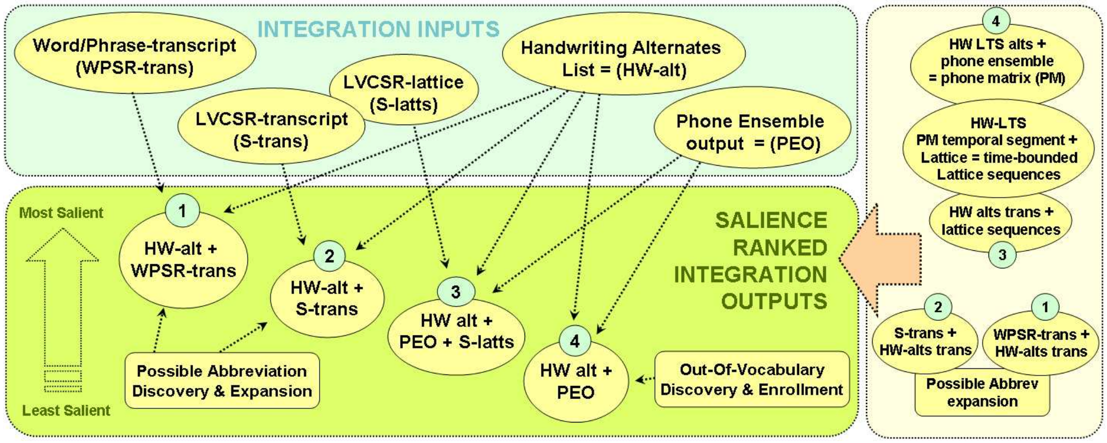

INTEGRATION INPUTS
Word/Phrase-transcript (WPSR-trans)
LVCSR-lattice (S-latts)
Handwriting Alternates List = (HW-alt)
LVCSR-transcript (S-trans)
Phone Ensemble output = (PEO)
Most Salient
Least Salient
1
HW-alt + WPSR-trans
2
HW-alt + S-trans
3
HW alt + PEO + S-latts
4
HW alt + PEO
SALIENCE RANKED INTEGRATION OUTPUTS
Possible Abbreviation Discovery & Expansion
Out-Of-Vocabulary Discovery & Enrollment
4
HW LTS alts + phone ensemble = phone matrix (PM)
HW-LTS PM temporal segment + Lattice = time-bounded Lattice sequences
HW alts trans + lattice sequences
3
2
S-trans + HW-alts trans
1
WPSR-trans + HW-alts trans
Possible Abbrev expansion

Figure 6.2: Salience rankings with flow lines indicating the input contributions to each. The right-side box offers a different view. (KEY: WPSR-trans = Word/Phrase-Spotting Recognizer transcript, S-trans = speech-transcript, S-latts = speech-lattices, HW-alt = handwriting alternates list, PEO = Phone Ensemble Output).

6.1.1 Saliency of Various Input Combination

Comparisons across the five information sources provide different perspectives on confidence. For example, the salience of each comparison is positioned graphically according to rank in Figure 6.2. The upper pane of the diagram in Figure 6.2 has the inputs, which are the same as those shown in Figure 6.1. The right pane of the diagram in Figure 6.2 shows how the inputs can be combined.

1. A matching relationship between word/phrase-spotter recognition transcripts (*WPSR-trans*) and handwriting alternates-list letter transcriptions (*HW-alts trans*), numbered *1*, is strong evidence for fusing speech/handwriting segment information; otherwise, the occurrence of the *WPSR-trans* can be used to expose possible handwritten abbreviations.

2. A matching relationship between speech transcripts (*S-trans*) and handwriting alternates-list letter transcriptions (*HW-alts trans*), numbered *2*, is again very strong evidence for fusing speech/handwriting segment information. It can also serve to identify handwritten abbreviations. For example, the handwritten letters, *IT*, when accompanied by the spoken phrase, "*Information Technology*," are strong evidence that *IT* should be interpreted as meaning *Information Technology.* In this case the handwriting and speech have a first-letter abbreviation relationship. A check for such standard relationships can confirm the abbreviation.

3. An aligned handwriting/phone-ensemble-output phone matrix (*PM*), which identifies the temporal segment corresponding to the handwriting, can be used to extract relevant word sequences from the speech transcriber's lattice. Those time-bounded *lattice sequences* can be combined with handwriting letter sequence alternatives (*HW alts trans*) to discover redundancy also (number *3*). In the saliency pane of Figure 6.2 the combination of inputs is labeled as *HW-alt + PEO + S-latts.*

4. The combination numbered *4*, between handwriting LTS alternates (*HW-LTS alts*) and phone ensemble outputs (*PEO*) exposes the handwriting segment location within accompanying speech on the basis of articulatory-feature based alignment of discovered redundancies. The phone sequences discovered in that alignment can be refined and yield a new word.

The lower pane in Figure 6.2 shows the relative saliency of various types of input matches listed above. An exact match between a handwriting alternative (*HW-alt*) and a Word/Phrase-Spotter Recognition (*WPSR-trans*) is very strong evidence for redundancy. Its salience is ranked *1.* For this event to occur, the word has to have been previously discovered as a redundancy, enrolled in the word phrase spotter, and then spoken again. An exact match between a handwriting alternative (*HW-alt*) and an utterance transcript term (*S-trans*) is also still strong evidence for accepting the transcript term. The salience of such a comparison is ranked *2* in Figure 6.2. Finally the occurrence of a match between handwriting (*HW-alt*) and speech recognizer lattice terms (*S-latts*) is considered more salient than matches of handwriting (*HW-alt*) and phone-ensemble-outputs (*PEO*). The least salient comparison type, between handwriting (*HW-alt*) and phone-ensemble-outputs (*PEO*), embodies the most uncertainty, which is gauged by the special measure of *coherence*, and thus also requires the most work in order to learn a new word.

These salience rankings are currently intuitive. They work reasonably well for proving the concept of basing dynamic learning of new vocabulary on the occurrence of multimodal redundancy, but SHACER will need a more stochastic, data-driven approach as more training data becomes available.

6.2 CHOOSING THE BEST COMBINATIONS OF INPUT INFORMATION

To combine information from all the different input sources SHACER currently follows the steps listed below:

1. Determine which handwriting alternative's letter-to-sound phone sequence (*HW-LTS alt*) is closest pronunciation-wise to the group of first-pass ensemble phone sequences. The result is the *1st-pass-closest-HW*.

2. Determine which (*HW-LTS alt*) is closest to the group of second-pass phone sequences. The result is the *2nd-pass-closest-HW*).

3. Compare, across both spelling and pronunciation, each pair of handwriting alternative and first-pass speech output, and then each pair of handwriting alternative and second-pass speech output. The scores of these pair-wise comparisons are a sum of the handwriting recognizer score, the speech recognizer score, and the normalized scores of the phone and letter alignment comparisons for each pair.

   During the final scoring of combined information sources, normalized phone alignment comparison scores are an average per-phone score based on the number of phones in the speech segment to which the handwriting is being compared. If all phones in the speech are matched then the score is 1.0 — a perfect match. An alignment with insertion and/or deletion errors will reduce the normalized match score. If there are more errors (e.g. substitutions, insertions or deletions) than correct matches then the normalized match score is 0. Normalized letter alignment comparisons are treated similarly.

4. If there exist combinations of handwriting alternatives and lattice word sequences, then those with (i) a high enough phone coherence, (ii) letter alignment score, and (iii) phone alignment score are examined and added to the list possible combinations.

There is no actual handwriting recognition score for word sequences extracted from the lattice. Thus handwriting recognition scores cannot be factored into the probability for such combinations. Thus the score of the lattice comparisons must be artificially scaled with respect to other combinations that do include handwriting recognition scores. Since the existence of high-scoring lattice sequences is on its own strong evidence of what the pronunciation should be, the lattice combinations are correspondingly biased to rank at the top of the list of comparisons across all information sources.

5. If, for a given utterance, there is a Word/Phrase-Spotter recognition then that is taken into account as strong evidence of what the spelling of the handwritten word should be.

6. If there is no Word/Phrase-Spotter recognition and no exact or near exact matches across the handwriting/lattice comparisons, then a determination from either (a) the handwriting/first-pass-speech, (b) handwriting/second-pass-speech or (c) handwriting/lattice comparisons is made as to what the most likely spelling and pronunciation ought to be. Information from the above *1st-pass-closest-HW* and *2nd-pass-closest-HW* is used in making this decision, to determine how much confidence to place in (a) and (b).

7. If any combination group ranks high enough compared to the others then its ranked and scored pairs are used to decide which handwriting alternative to use as a basis for the spelling.

8. If no combination group ranks high enough then all combinations are sorted and the best scoring pair becomes the basis for creating the list of alternate spellings and pronunciations.

The final result of this alignment-based integration process are output messages from the master speech recognizer, like those shown in Figure 6.3. The *score* listed for these examples is actually that of the most coherent alignment matrix for the redundant inputs involved in the new word discovery. These messages are routed back to the multiparser for both persistent enrollment in the learning accumulator structures and immediate inclusion in the displayed Gantt chart.

Ultimately, with a larger database for training, SHACER will employ stochastic pattern recognition approaches — like neural nets, maximum entropy models or conditional

```
Upper Example:
    score:
    ------------
    0.742,

    spelling        pronunciation
    --------------  -------------------------
    'Buy Computer'  'b ay k ah m p y uw t er'

Lower Example:
    score:
    ------------
    0.725,

    spelling        pronunciation
    --------------  -------------------------
    'Joe Browning'  'jh ow b r aw n ih ng'
```

Figure 6.3: Example responses to the multiparser's request for phonetic alignment and redundancy discovery.

random fields — to model the combination of variously weighted information under all the various conditions of integration. We believe that having identified the heuristic conditions described above for determining what combination factors should be taken into account will eventually help in designing these stochastic approaches to make better combinatory choices.

6.3 INTEGRATION EXAMPLES

In the remainder of this chapter we examine in detail examples of handwriting and speech integration, which show how SHACER's approach can yield substantial improvements in recognition. In each example we highlight how the refined pronunciation of a multimodally redundant term is used as an integration decision metric, to assign confidence to the various input information sources. All of the examples are based on the second of the *G* series of meetings collected as part of the CALO project at SRI. This meeting is referred to as the *G2* meeting. The whiteboard ink that occurred during this meeting is shown in Figure 6.4. The processed understanding of that ink is shown in Appendix D.

To process this *G2* meeting the recorded ink and speech were played back in appropriate order within SHACER's MultiPlayer Suite for off-line analysis and integration. All examples discussed result from this off-line playback analysis. Previously logged messages from multiple input streams are processed in lock-step mode, which guarantees that each input is fully processed before the next input is sent. This is necessary because some of

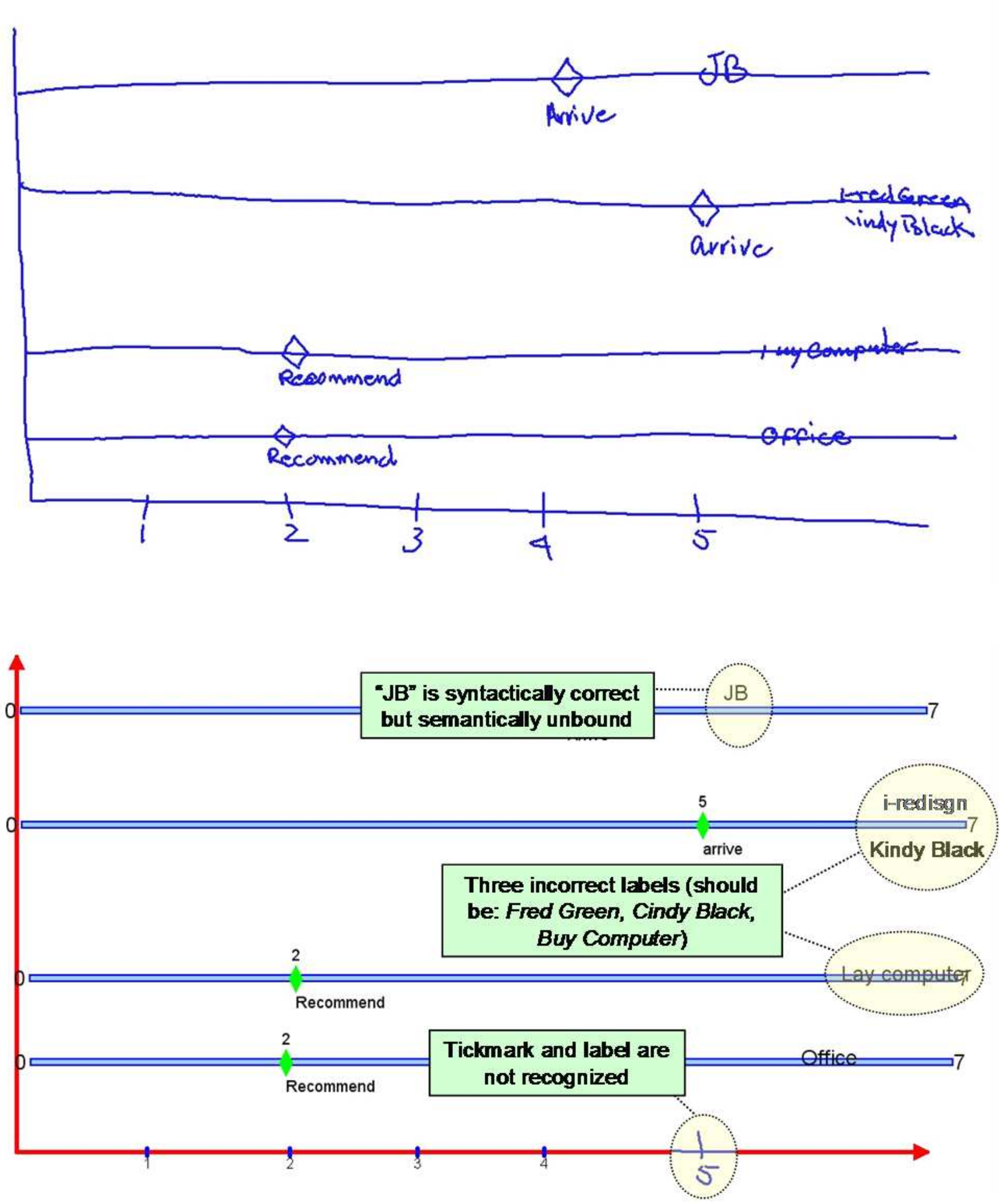
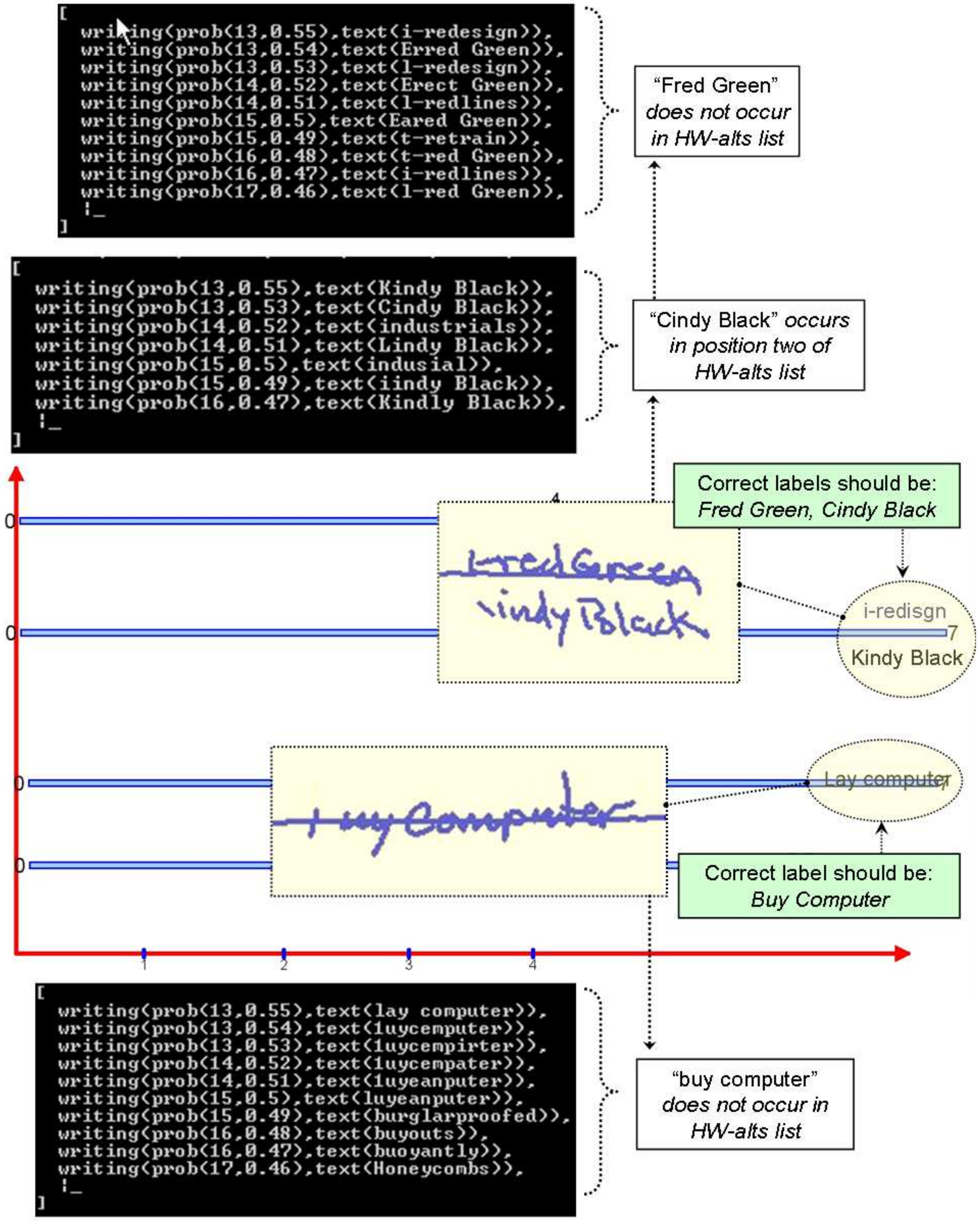

JB
Arrive
Fred Green
Cindy Black
arrive
Buy Computer
Recommend
Office
Recommend
1
2
3
4
5
0
"JB" is syntactically correct but semantically unbound
JB
7
0
5
arrive
i-redisgn
Kindy Black
7
Three incorrect labels (should be: *Fred Green, Cindy Black, Buy Computer*)
2
Lay computer
Recommend
2
Recommend
Tickmark and label are not recognized
Office
7
1
2
3
4

Figure 6.4: *Upper Diagram.* The ink input for meeting *G2*. In this meeting the arrival times for three new hires are scheduled. The new hires are named Joe Browning (*JB*), *Fred Green* and *Cindy Black*. Along with their arrival times, recommendations for their equipment and office space are also scheduled. *Lower Diagram.* *G2* meeting analysis errors when SHACER is not used. Top: missing semantics for the abbreviation, *JB*. Middle: three misspelled constituent labels due to incorrect handwriting recognition. Bottom: unrecognized tick mark and label due to sketch mis-segmentation.



```
[
  writing(prob(13,0.55),text(i-redesign)),
  writing(prob(13,0.54),text(Erred Green)),
  writing(prob(13,0.53),text(l-redesign)),
  writing(prob(14,0.52),text(Erect Green)),
  writing(prob(14,0.51),text(l-redlines)),
  writing(prob(15,0.5),text(Eared Green)),
  writing(prob(15,0.49),text(t-retrain)),
  writing(prob(16,0.48),text(t-red Green)),
  writing(prob(16,0.47),text(i-redlines)),
  writing(prob(17,0.46),text(l-red Green)),
  !_
]
```

"Fred Green" *does not occur in HW-alts list*

```
[
  writing(prob(13,0.55),text(Kindy Black)),
  writing(prob(13,0.53),text(Cindy Black)),
  writing(prob(14,0.52),text(industrials)),
  writing(prob(14,0.51),text(Lindy Black)),
  writing(prob(15,0.5),text(indusial)),
  writing(prob(15,0.49),text(iindy Black)),
  writing(prob(16,0.47),text(Kindly Black)),
  !_
]
```

"Cindy Black" *occurs in position two of HW-alts list*

Correct labels should be: *Fred Green, Cindy Black*

Correct label should be: *Buy Computer*

```
[
  writing(prob(13,0.55),text(lay computer)),
  writing(prob(13,0.54),text(1uycemputer)),
  writing(prob(13,0.53),text(1uycempirter)),
  writing(prob(14,0.52),text(1uycempater)),
  writing(prob(14,0.51),text(1uyeanputer)),
  writing(prob(15,0.5),text(luyeanputer)),
  writing(prob(15,0.49),text(burglarproofed)),
  writing(prob(16,0.48),text(buyouts)),
  writing(prob(16,0.47),text(buoyantly)),
  writing(prob(17,0.46),text(Honeycombs)),
  !_
]
```

"buy computer" *does not occur in HW-alts list*

Figure 6.5: For the three incorrectly interpreted label names during the *G2* meeting, correct handwriting interpretation was either missing from the list of recognition alternates or was not the first alternate listed.

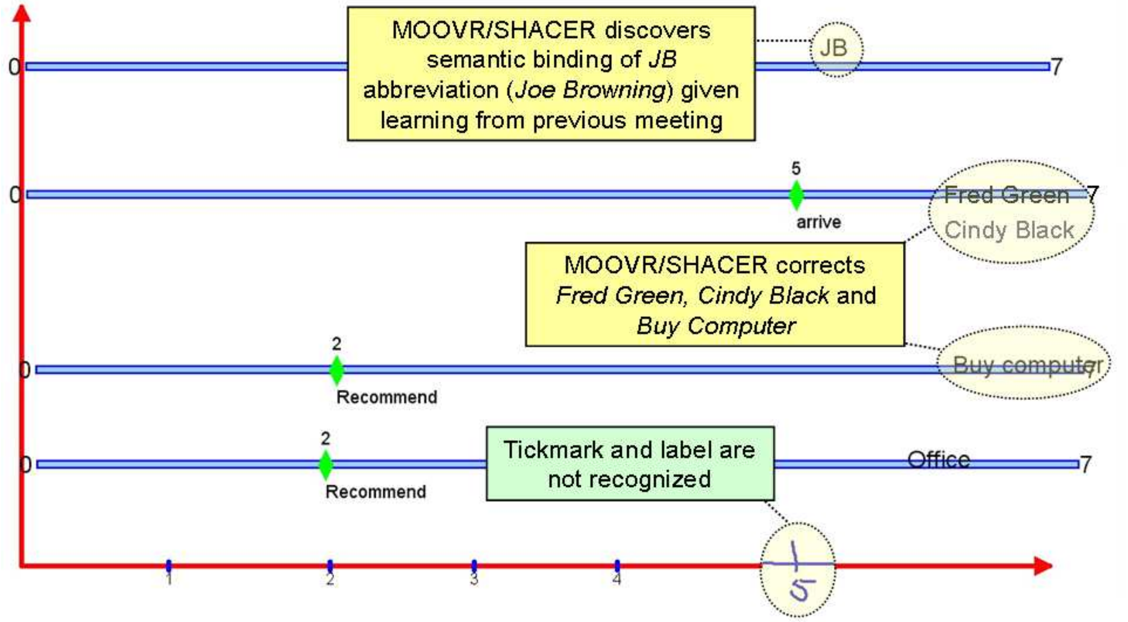

MOOVR/SHACER discovers semantic binding of *JB* abbreviation (*Joe Browning*) given learning from previous meeting
JB
0
7
5
arrive
Fred Green
Cindy Black
MOOVR/SHACER corrects *Fred Green, Cindy Black* and *Buy Computer*
2
Recommend
Buy computer
Tickmark and label are not recognized
Office
1
2
3
4
1/5

Figure 6.6: *G2* meeting analysis corrections when SHACER is used. Top: abbreviation semantics discovered (*JB* = *Joe Browning*). Middle: three constituent labels dynamically enrolled in WPSR with correct spelling, semantics and pronunciation. Bottom: Unrecognized tickmark and label not processed by SHACER at this time, still incorrect.

SHACER's recognizers are slower than realtime. The errors that occur in this Multiplayer analysis for meeting G2, when SHACER is not used, are depicted in Figure 6.4, and Figure 6.5 shows that in each case the correct handwriting interpretation was either missing from the list of recognition alternates or was not the first alternate listed.

During the *G2* meeting each of the handwriting events was accompanied by redundant speech. So, the mistaken interpretations could be corrected by SHACER's integration of information from both input streams. Figure 6.6 shows that by using SHACER four of the five errors were corrected. In the remaining sections we will show how SHACER integrated information to make these corrections.

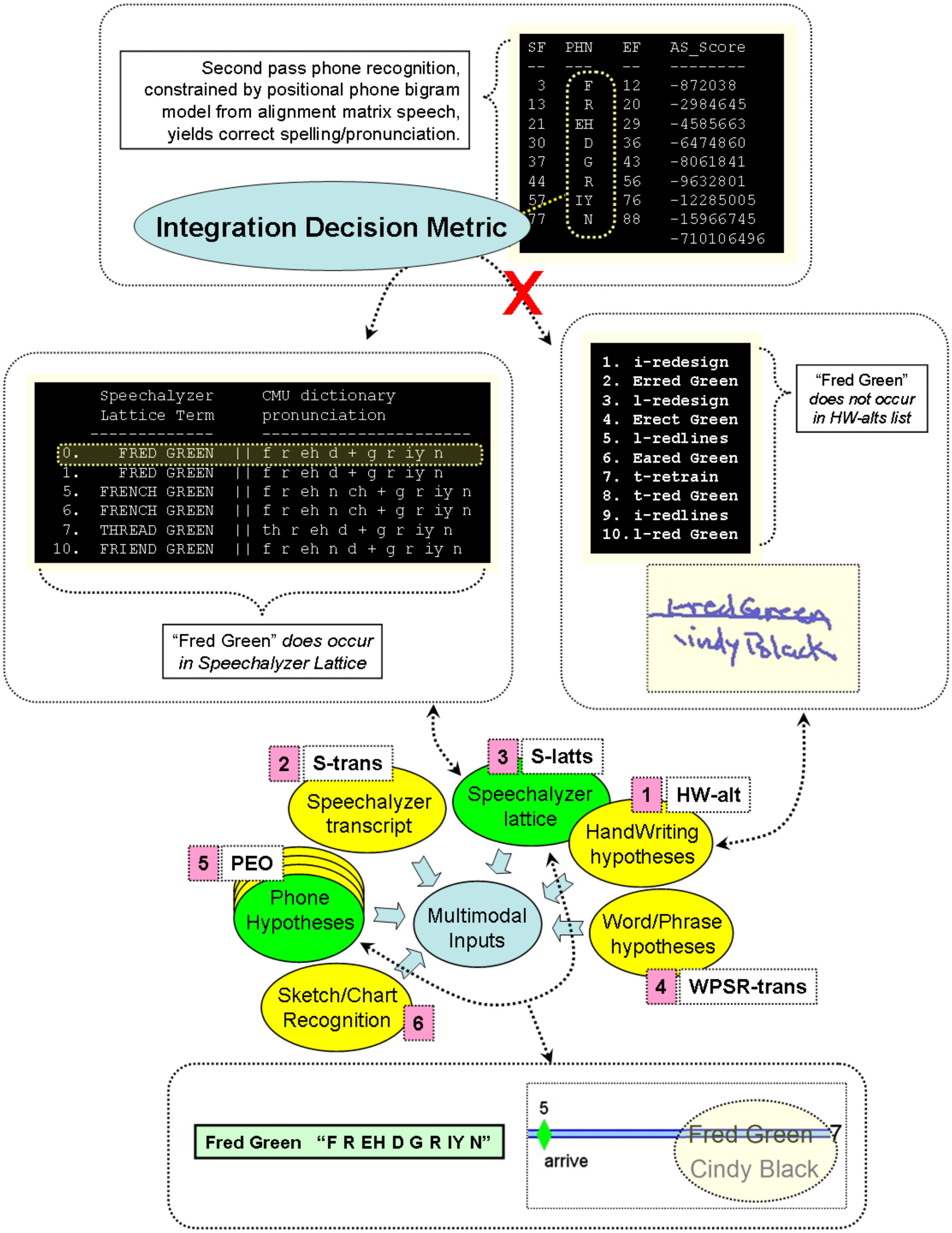
6.3.1 Lattice Alignment Fusion: 'Fred Green'

In finding the correct spelling and pronunciation for *Fred Green* (as illustrated in Figure 6.7), salience-combination *3* is the critical piece of information. This salience combination is shown in Figure 6.2's saliency pane as the third combination of inputs labeled as *HW-alt + PEO + S-latts*.

In this example, *Fred Green* did not exist in the handwriting recognizer's alternates list (*HW-alts*). However, after aligning *HW-alts* with the phone-ensemble outputs (*PEOs*), there was enough phonetic information from these sources to make a coherent alignment, indicating a possible redundancy. This alignment, which is not shown in Figure 6.7, also provided estimates of the temporal boundaries of the redundancy within the spoken utterance. Using the positional phone bigram model, which was dynamically built from the alignment matrix over the redundant segment, second-pass phonetic recognition yielded a refined pronunciation. This refined pronunciation is shown at the top of Figure 6.7 — set off with a dotted highlight. This refined pronunciation was then used as an *integration decision metric* against which alternate spelling and pronunciation choices from (a) the speech transcripts, (b) speech lattices and (c) word/phrase-spotter recognitions, and (d) handwriting information were compared.

The temporal segmentation boundaries from the aligned handwriting/phone-ensemble-output matrix were used to extract word sequences from the Speechalyzer lattice. The list of extracted word sequences is shown in Figure 6.7. The best scoring extracted word sequence was an exact pronunciation match for the refined second-pass phone sequence output (e.g. *F R EH D G R IY N = f r eh d g r iy n*). This exact match was better than any competing match with the handwriting alternates list. Thus the comparison with the handwriting alternates list is marked with red *X* indicating that it was not chosen as being closest to the *integration decision metric*. The speech transcript, which is not shown in Figure 6.7, in this case it was *Fred's Green*. The transcriber inserted a possessive *'s*, because the two-word proper name sequence had no likelihood in the speech transcribing recognizer's language model. The match between the *integration decision metric* and this transcription's pronunciation (*F R EH D Z G R IY N*) was not as good as the match with the best lattice extraction. For this utterance there were no previously enrolled words recognized by the word/phrase-spotter, so there were no comparisons made in that regard.

The multimodal inputs graphic in the lower part of Figure 6.7 shows that the critical participants in this integration decision were the phone ensemble output (*PEO*) matrix with aligned handwriting, together with information extracted from the Speechalyzer lattice. The bottom pane of Figure 6.7 shows that in this instance the correct spelling, pronunciation and semantics were discovered even though they did not exist in the outputs of the speech transcriber and handwriting recognizer. Once such a label has been recovered it is enrolled in the Word/Phrase-Spotting Recognizer, and can thus be more readily recognized when it is next uttered.



Second pass phone recognition, constrained by positional phone bigram model from alignment matrix speech, yields correct spelling/pronunciation.

```
SF  PHN   EF   AS_Score
--  ---   --   --------
 3    F   12   -872038
13    R   20   -2984645
21   EH   29   -4585663
30    D   36   -6474860
37    G   43   -8061841
44    R   56   -9632801
57   IY   76   -12285005
77    N   88   -15966745
               -710106496
```

```
      Speechalyzer       CMU dictionary
      Lattice Term       pronunciation
    -------------        -----------------------
 0.    FRED GREEN   || f r eh d + g r iy n
 1.    FRED GREEN   || f r eh d + g r iy n
 5.  FRENCH GREEN   || f r eh n ch + g r iy n
 6.  FRENCH GREEN   || f r eh n ch + g r iy n
 7.  THREAD GREEN   || th r eh d + g r iy n
10.  FRIEND GREEN   || f r eh n d + g r iy n
```

"Fred Green" *does occur in Speechalyzer Lattice*

```
1.  i-redesign
2.  Erred Green
3.  l-redesign
4.  Erect Green
5.  l-redlines
6.  Eared Green
7.  t-retrain
8.  t-red Green
9.  i-redlines
10. l-red Green
```

"Fred Green" *does not occur in HW-alts list*

Figure 6.7: Discovering the correct spelling, semantics and pronunciation of Fred Green by aligning and fusing speech and handwriting information sources.

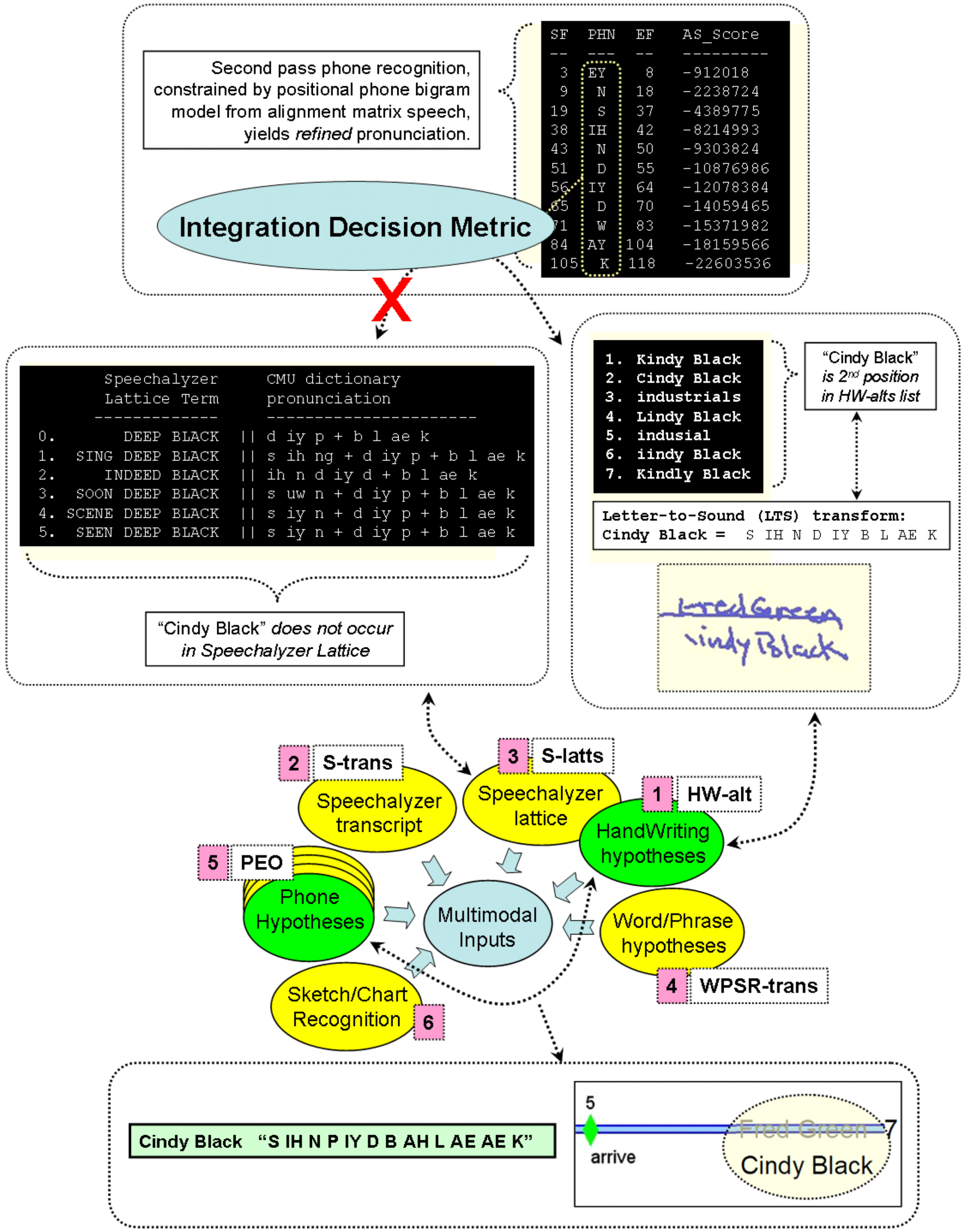
6.3.2 Speech/Handwriting Fusion: 'Cindy Black'

In finding the correct spelling of *Cindy Black* (as illustrated in Figure 6.8), salience-combination *4* is the critical piece of information. This salience combination is shown in Figure 6.2's saliency pane as the fourth combination of inputs labeled as *HW-alt + PEO*.

In this case the correct term sequence did not occur in either the Speechalyzer transcript or lattice. The transcript, *Cindy's Black*, which is not shown in Figure 6.8, had an inserted possessive *'s*, because again the two-word proper name sequence had no likelihood in the speech transcribing recognizer's language model. As with *Fred Green* this meant that *Cindy Black* was an OOV term. However, *Cindy Black* did occur as the second hypothesis on the handwriting alternates list (*HW-alts*). The letter-to-sound (LTS) transformation for that alternative is shown Figure 6.8.

In comparing the *integration decision metric* to the various inputs, the best pronunciation match was actually one of the first-pass ensemble phone pronunciations, which ultimately became the pronunciation shown in Figure 6.8's bottom pane. Since a phone ensemble pronunciation was the best match, the rest of the integration decision was made by scoring all combinations of spellings, spelling LTS transformations, pronunciations and pronunciation STL (sound-to-letter) transformations. The score of such tuples depends on the acoustic score of the phone ensemble pronunciations, the handwriting recognizer score of the spellings, and the normalized articulatory-feature based distance between the pronunciation/spelling-LTS combinations and spelling/pronunciation-STL combinations for the tuple. The best scoring tuple's spelling was that of the second handwriting alternative, *Cindy Black*. Thus, the comparison of the *integration decision metric* to the lattice is marked with red *X* in Figure 6.8 because it was not the most useful comparison in this case. The best matching pronunciation was that of the phone-ensemble pronunciation that was closest to the *integration decision metric*.

The canonical pronunciation for *Cindy Black* was not among the results of either the ensemble speech recognition, the second pass search, or the integration step. However, the phone ensemble pronunciation, which was chosen by the *integration decision metric* (shown in the bottom pane of Figure 6.8) did show intriguing evidence of phonetic adaptation. For example, there is common tendency to say words like *black* as two syllables — *bah-lack* — instead of one. Instead of being a pronunciation recognition error this non-canonical pronunciation could represent pronunciation adaptation. If that is so, then after it was enrolled in the word/phrase-spotter it would aid in making the term more recognizable when next it was uttered.



```
SF  PHN  EF    AS_Score
--  ---  --    ---------
 3  EY    8    -912018
 9   N   18    -2238724
19   S   37    -4389775
38  IH   42    -8214993
43   N   50    -9303824
51   D   55    -10876986
56  IY   64    -12078384
65   D   70    -14059465
71   W   83    -15371982
84  AY  104    -18159566
105  K  118     -22603536
```

```
         Speechalyzer     CMU dictionary
         Lattice Term     pronunciation
        -------------     ----------------------
0.         DEEP BLACK  || d iy p + b l ae k
1.    SING DEEP BLACK  || s ih ng + d iy p + b l ae k
2.      INDEED BLACK   || ih n d iy d + b l ae k
3.    SOON DEEP BLACK  || s uw n + d iy p + b l ae k
4.   SCENE DEEP BLACK  || s iy n + d iy p + b l ae k
5.    SEEN DEEP BLACK  || s iy n + d iy p + b l ae k
```

```
1.  Kindy Black
2.  Cindy Black
3.  industrials
4.  Lindy Black
5.  indusial
6.  iindy Black
7.  Kindly Black
```

```
Letter-to-Sound (LTS) transform:
Cindy Black =   S IH N D IY B L AE K
```

Second pass phone recognition, constrained by positional phone bigram model from alignment matrix speech, yields *refined* pronunciation.
Integration Decision Metric
"Cindy Black" *does not occur in Speechalyzer Lattice*
"Cindy Black" *is 2nd position in HW-alts list*
Fred Green
Cindy Black
2 S-trans
Speechalyzer transcript
3 S-latts
Speechalyzer lattice
1 HW-alt
HandWriting hypotheses
5 PEO
Phone Hypotheses
Multimodal Inputs
Word/Phrase hypotheses
4 WPSR-trans
Sketch/Chart Recognition
6
Cindy Black "S IH N P IY D B AH L AE AE K"
5
arrive
Fred Green
7
Cindy Black

Figure 6.8: Discovering the correct spelling for Cindy Black and introducing the possibility of pronunciation adaptation.

The multimodal inputs graphic in the lower part of Figure 6.8 shows that the critical participants in this integration decision were the phone ensemble output (*PEO*) matrix with aligned handwriting, together with handwriting alternates list (*HW-alts*). The bottom pane of Figure 6.8 shows that in this instance the correct spelling and semantics were again discovered even though they did not exist in the output of the speech transcriber and were not the first alternative in the handwriting recognizer's output.

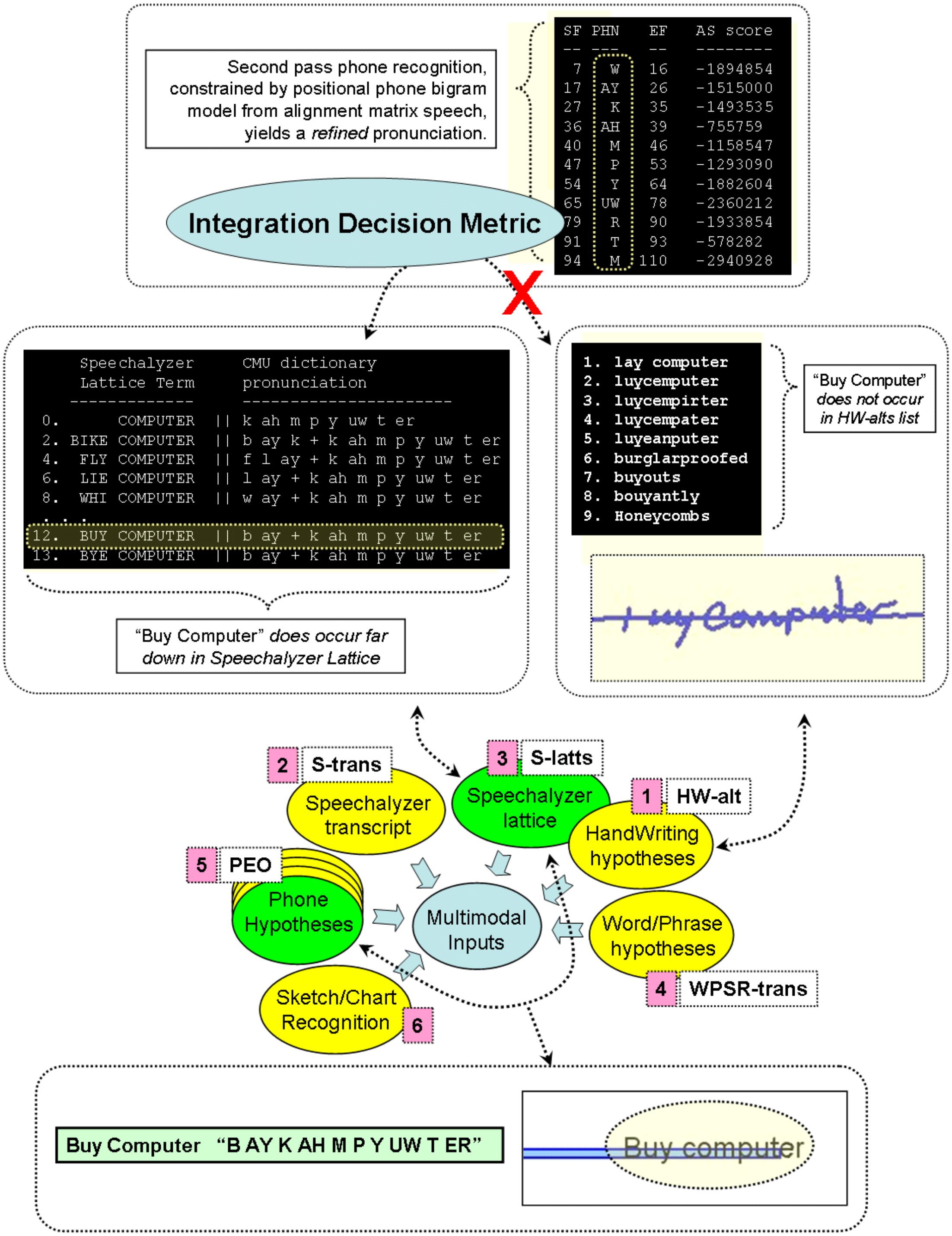
6.3.3 Speech/Handwriting Fusion: 'Buy Computer'

In finding the correct spelling of *Buy Computer* (as illustrated in Figure 6.9), SHACER was again able to leverage the refined pronunciation produced by the second-pass phone recognition over cached speech features as an anchor for comparison. Comparing the term sequences extracted from the Speechalyzer lattice to the *integration decision metric* yielded a closest match for the 13th alternative — very low on the alternates list by virtue of its acoustic and language model scores from the transcribing speech recognizer. This strong comparative match boosts it to have the best combined score, and thus allows SHACER to recover the correct spelling and pronunciation in this instance.

The multimodal inputs graphic in the lower part of Figure 6.9 shows that the critical participants in this integration decision were the phone ensemble output (*PEO*) matrix with aligned handwriting, together with information extracted from the Speechalyzer lattice. Although the lattice information is much less prominent than in the *Fred Green* case it is nonetheless sufficient for correct integration.

6.3.4 Speech/Handwriting Fusion: Discussion

In summary it seems clear that the array of evidence (e.g., ensemble speech, handwriting, Speechalyzer transcripts and lattices, WPSR recognition) that we have at our disposal is very rich, and provides a basis for making many reasonable recognition choices in context. In Chapter 7 we will show test results on the entire development test set from which these examples are drawn as well as results on a separate held-out test set of meeting data. As those test results indicate SHACER's integration of redundant multimodal inputs already can make a significant difference in understanding; still, we believe that we are just beginning to explore the types of features available in this space and the ways in which we can take advantage of this rich information across redundant modal inputs.



Second pass phone recognition, constrained by positional phone bigram model from alignment matrix speech, yields a *refined* pronunciation.

```
SF PHN   EF   AS score
-- ---   --   --------
 7   W   16   -1894854
17  AY   26   -1515000
27   K   35   -1493535
36  AH   39   -755759
40   M   46   -1158547
47   P   53   -1293090
54   Y   64   -1882604
65  UW   78   -2360212
79   R   90   -1933854
91   T   93   -578282
94   M  110   -2940928
```

```
    Speechalyzer     CMU dictionary
    Lattice Term     pronunciation
   -------------     ----------------------
 0.      COMPUTER || k ah m p y uw t er
 2. BIKE COMPUTER || b ay k + k ah m p y uw t er
 4.  FLY COMPUTER || f l ay + k ah m p y uw t er
 6.  LIE COMPUTER || l ay + k ah m p y uw t er
 8.  WHI COMPUTER || w ay + k ah m p y uw t er
 . . .
12.  BUY COMPUTER || b ay + k ah m p y uw t er
13.  BYE COMPUTER || b ay + k ah m p y uw t er
```

"Buy Computer" *does occur far down in Speechalyzer Lattice*

```
1. lay computer
2. luycemputer
3. luycempirter
4. luycempater
5. luyeanputer
6. burglarproofed
7. buyouts
8. bouyantly
9. Honeycombs
```

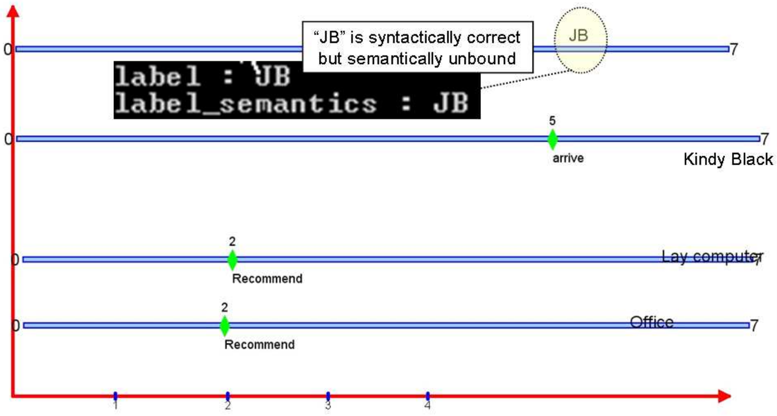
"Buy Computer" *does not occur in HW-alts list*

Buy Computer "B AY K AH M P Y UW T ER"

Figure 6.9: Discovering the correct spelling and pronunciation of the taskline label, *buy computer*.

6.4 DISCOVERING HANDWRITING ABBREVIATION SEMANTICS

In Figure 6.10 below the handwritten abbreviation, *JB*, is syntactically correct but semantically unbound. The system only knows that the symbol *JB* is a handwritten taskline label, but is has no idea that *JB* may be an abbreviation with larger meaning.

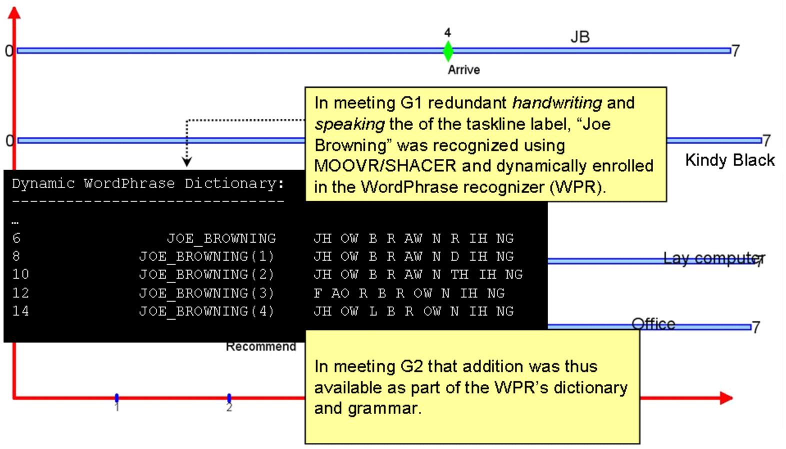


```
label : JB
label_semantics : JB
```

Figure 6.10: Unbound semantics of taskline label, *JB*.

6.4.1 The Role of the Word/Phrase-Spotting Recognizer

Without SHACER the system does not know that *JB* has a broader sphere of reference, and indeed shares the same meaning as the spoken and handwritten term, *Joe Browning*. SHACER has the capability to make this discovery and to do it dynamically based on Word/Phrase-Spotting recognition (WPSR) enrollments from the previous meeting, G1 (as shown in Figure 6.11). WPSR acts a persistent store of enrolled spelling/pronunciation combinations, which is cumulative either within a single meeting or across a series of meetings, and thus supports boot-strapped recognition improvements as the system is used over time. Combining handwriting alternates (*HW-alts*) with word/phrase-spotting recognition transcripts (*WPSR-trans*) in the most salient of the combination types shown in Figure 6.2. A fully handwritten term, which has already been learned and enrolled by the system, occurring also as a WPSR transcript is very strong evidence of redundancy. So strong, in fact, that it can also be used to associate larger meanings with abbreviations.



```
Dynamic WordPhrase Dictionary:
------------------------------
…
6              JOE_BROWNING    JH OW B R AW N R IH NG
8           JOE_BROWNING(1)    JH OW B R AW N D IH NG
10          JOE_BROWNING(2)    JH OW B R AW N TH IH NG
12          JOE_BROWNING(3)    F AO R B R OW N IH NG
14          JOE_BROWNING(4)    JH OW L B R OW N IH NG
```

In meeting G1 redundant *handwriting* and *speaking* the of the taskline label, "Joe Browning" was recognized using MOOVR/SHACER and dynamically enrolled in the WordPhrase recognizer (WPR).

In meeting G2 that addition was thus available as part of the WPR's dictionary and grammar.

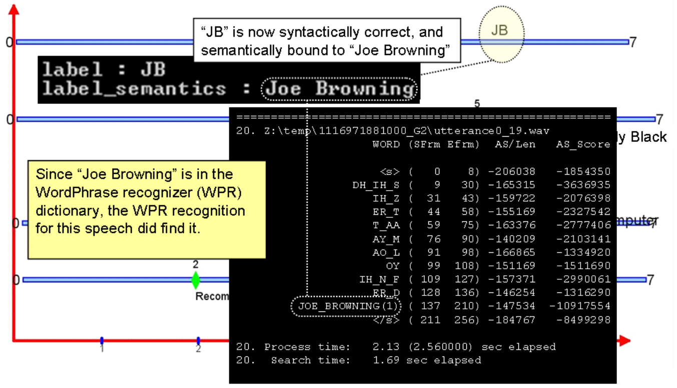
Figure 6.11: Word/Phrase-Spotting Recognizer (WPSR) acting as a persistent store of spelling and pronunciations across a series of meetings (in this case from meeting G1 to meeting G2).

In meeting G2 as the user wrote *JB* he also said, "This is our timeline for *Joe Browning*." The Speechalyzer recognition for this utterance was, "This is our timeline for *job running*," because *Browning* was not in the Speechalyzer dictionary. However, since *Joe Browning* was enrolled in the WPSR by SHACER during meeting G1, it was recognized by WPSR in meeting G2 for this user utterance, and this recognition provided the basis for binding *JB* to *Joe Browning* as depicted in Figure 6.12.

6.4.2 Comparing Word/Phrase-Spotter Recognition to Handwriting

Term recognition in WPSR is first used to match to the handwriting alternates list (*HW-alts*)), as shown in the upper pane of Figure 6.13. If the bounds of the alignment section for the *HW-alts* are significantly different than those of the WPSR term transcript, then this difference can be used to expose the existence of a handwritten abbreviation. This situation is shown in Figure 6.13. The HW abbreviation phone sequence hypotheses cover a segment across the ensemble speech phone sequence alignment much shorter than the bounds of the WPSR term's end boundary (e.g. the significant difference between frame *210* and frame *160* shown in Figure 6.13). This significant difference triggers a decision to explore this WPSR recognition event (e.g. *Joe Browning*) as the semantics of the handwritten abbreviation.

SHACER uses two pieces of evidence to make the final decision on binding the handwritten abbreviation. First the distance of each handwritten hypothesis from the WPSR



"JB" is now syntactically correct, and semantically bound to "Joe Browning"

```
label : JB
label_semantics : Joe Browning
```

Since "Joe Browning" is in the WordPhrase recognizer (WPR) dictionary, the WPR recognition for this speech did find it.

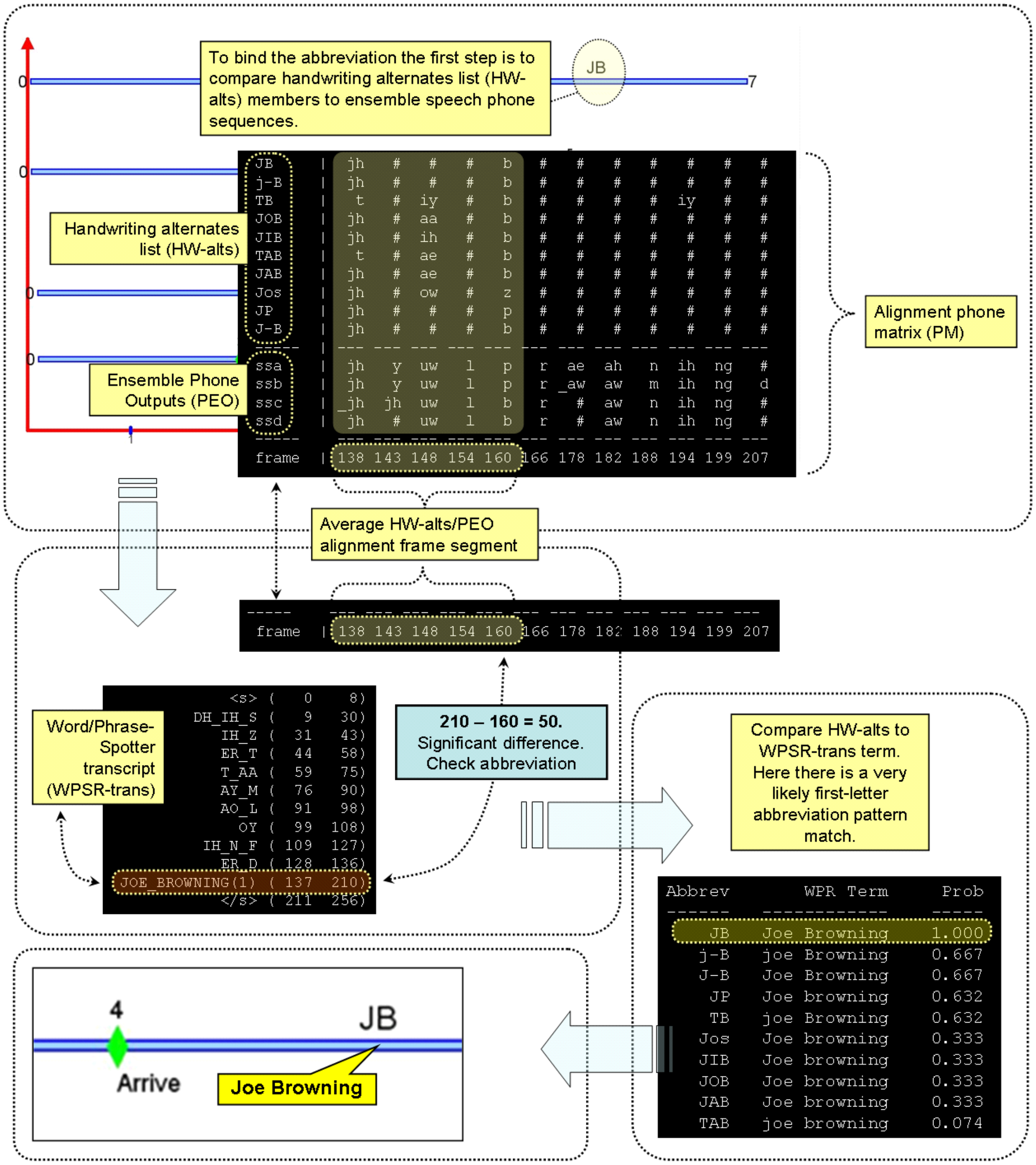
```
=============================================================
20. Z:\temp\1116971881000_G2\utterance0_19.wav
                 WORD (SFrm Efrm)  AS/Len   AS_Score

                  <s> (   0    8) -206038   -1854350
             DH_IH_S (   9   30) -165315   -3636935
                IH_Z (  31   43) -159722   -2076398
                ER_T (  44   58) -155169   -2327542
                T_AA (  59   75) -163376   -2777406
                AY_M (  76   90) -140209   -2103141
                AO_L (  91   98) -166865   -1334920
                  OY (  99  108) -151169   -1511690
              IH_N_F ( 109  127) -157371   -2990061
                ER_D ( 128  136) -146254   -1316290
      JOE_BROWNING(1) ( 137  210) -147534  -10917554
                 </s> ( 211  256) -184767   -8499298

20. Process time:   2.13 (2.560000) sec elapsed
20.  Search time:   1.69 sec elapsed
```

Figure 6.12: Word/Phrase-Spotting Recognition of *Joe Browning* as the basis of binding the semantics of the handwritten abbreviation *JB*.

output is measured. At the time this example was created SHACER only considered the handwriting hypotheses as potential first letter abbreviations. Now it also considers prefix abbreviations, and in the future this check will be expanded to consider other varieties of abbreviation templates. This measurement yields a highly likely first letter abbreviation interpretation (Figure 6.13).

The second piece of evidence SHACER uses is to examine the Speechalyzer lattice for term sequences spanning the boundaries found in the WPSR recognition transcript. Those lattice sequences that are close enough to the WPSR transcript are aligned with the phone ensemble outputs. This alignment is used to create a positional phone bigram model, which in turn constrains a second-pass phone recognition. The refined pronunciation from that second pass output is then the *integration decision metric* as shown in Figure 6.14. The check between the *integration decision metric* and the WPSR pronunciation confirms that the WPSR recognition was not spurious.

After these two comparisons, if the first letter abbreviation distance is close enough and there is a sufficient match between WPSR output and the refined pronunciation, then SHACER decides to treat the handwritten term as an abbreviation, and binds the WPSR proper name semantics to it. Figure 6.14 shows the result of the second-pass recognition, a plausible pronunciation adaptation for the term Joe Browning, which in turn is added back into the WPSR as another pronunciation alternative. In the future we will use such additions to refine the WPSR pronunciation alternatives (using clustering and centroid


0
0
0
0
1
7
JB
To bind the abbreviation the first step is to compare handwriting alternates list (HW-alts) members to ensemble speech phone sequences.
Handwriting alternates list (HW-alts)
Ensemble Phone Outputs (PEO)
Alignment phone matrix (PM)
Average HW-alts/PEO alignment frame segment
Word/Phrase-Spotter transcript (WPSR-trans)
210 – 160 = 50. Significant difference. Check abbreviation
Compare HW-alts to WPSR-trans term. Here there is a very likely first-letter abbreviation pattern match.
4
Arrive
JB
Joe Browning

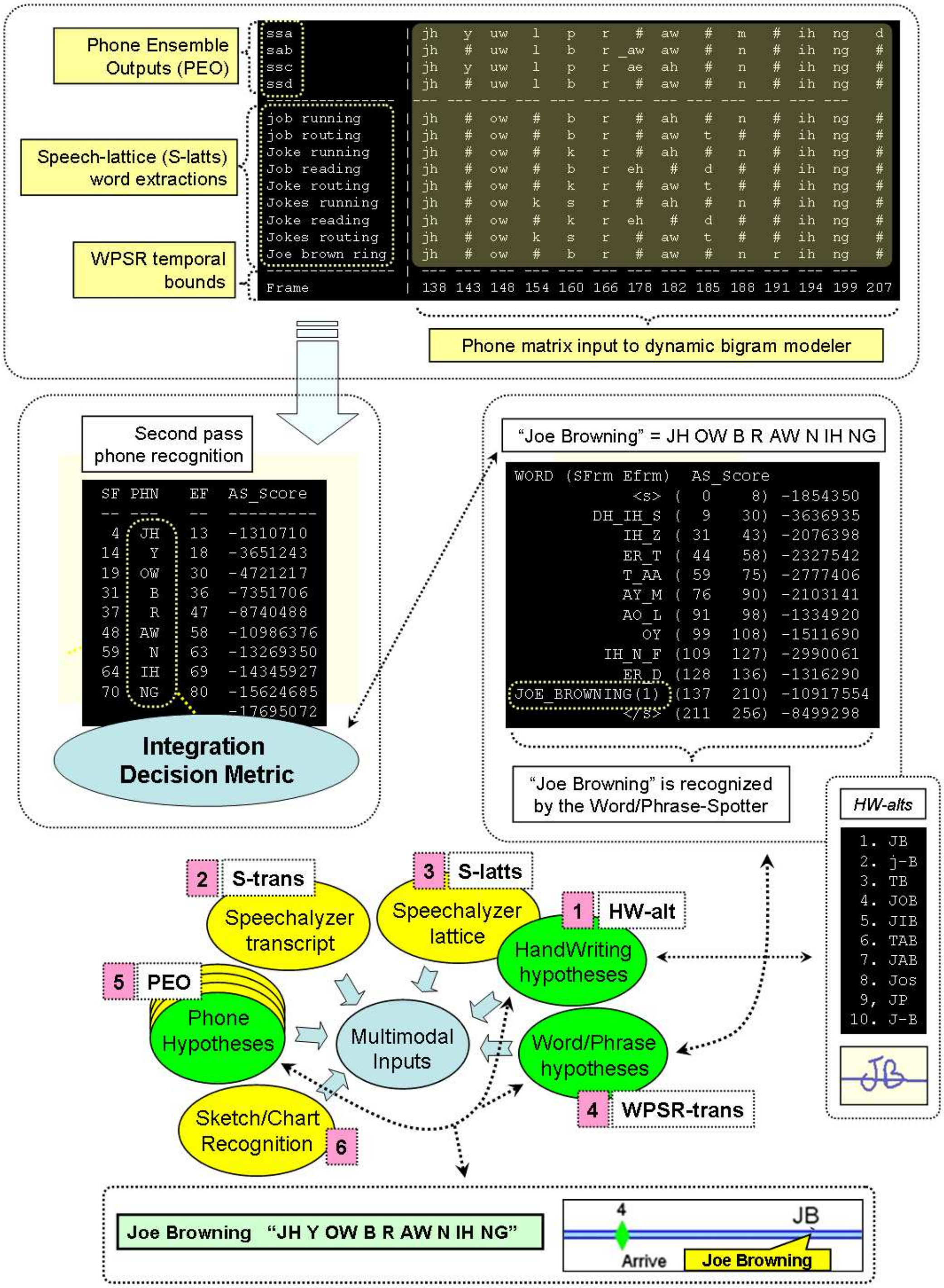
```
JB      |  jh   #   #   #   b   #   #   #   #   #   #   #
j-B     |  jh   #   #   #   b   #   #   #   #   #   #   #
TB      |   t   #  iy   #   b   #   #   #   #  iy   #   #
JOB     |  jh   #  aa   #   b   #   #   #   #   #   #   #
JIB     |  jh   #  ih   #   b   #   #   #   #   #   #   #
TAB     |   t   #  ae   #   b   #   #   #   #   #   #   #
JAB     |  jh   #  ae   #   b   #   #   #   #   #   #   #
Jos     |  jh   #  ow   #   z   #   #   #   #   #   #   #
JP      |  jh   #   #   #   p   #   #   #   #   #   #   #
J-B     |  jh   #   #   #   b   #   #   #   #   #   #   #
-----   | --- --- --- --- --- --- --- --- --- --- --- ---
ssa     |  jh   y  uw   l   p   r  ae  ah   n  ih  ng   #
ssb     |  jh   y  uw   l   p   r _aw  aw   m  ih  ng   d
ssc     | _jh  jh  uw   l   b   r   #  aw   n  ih  ng   #
ssd     |  jh   #  uw   l   b   r   #  aw   n  ih  ng   #
-----     --- --- --- --- --- --- --- --- --- --- --- ---
frame   | 138 143 148 154 160 166 178 182 188 194 199 207
```

```
-----     --- --- --- --- --- --- --- --- --- --- --- ---
frame   | 138 143 148 154 160 166 178 182 188 194 199 207
```

```
          <s> (   0   8)
    DH_IH_S (   9   30)
       IH_Z (  31   43)
       ER_T (  44   58)
       T_AA (  59   75)
       AY_M (  76   90)
       AO_L (  91   98)
         OY (  99  108)
     IH_N_F ( 109  127)
       ER_D ( 128  136)
JOE_BROWNING(1) ( 137  210)
        </s> ( 211  256)
```

```
Abbrev       WPR Term       Prob
------    ------------    -----
   JB     Joe Browning    1.000
  j-B     joe Browning    0.667
  J-B     Joe Browning    0.667
   JP     Joe browning    0.632
   TB     joe Browning    0.632
  Jos     Joe browning    0.333
  JIB     Joe browning    0.333
  JOB     Joe browning    0.333
  JAB     Joe browning    0.333
  TAB     joe browning    0.074
```

Figure 6.13: Upper pane: phonetic alignment of phone ensemble outputs with handwriting hypotheses, for the handwritten abbreviation, *JB*, and the section of the utterance, "This is our timeline for *Joe Browning*," that corresponds to *Joe Browning*. The alignment mismatch in temporal boundaries, triggers a check for a possible abbreviation. In this case there is a strong first-letter abbreviation match.



```
ssa                |  jh   y  uw   l   p   r   #  aw   #   m   #  ih  ng   d
sab                |  jh   #  uw   l   b   r _aw  aw   #   n   #  ih  ng   #
ssc                |  jh   y  uw   l   p   r  ae  ah   #   n   #  ih  ng   #
ssd                |  jh   #  uw   l   b   r   #  aw   #   n   #  ih  ng   #
---------------       --- --- --- --- --- --- --- --- --- --- --- --- ---
job running        |  jh   #  ow   #   b   r   #  ah   #   n   #  ih  ng   #
job routing        |  jh   #  ow   #   b   r   #  aw   t   #   #  ih  ng   #
Joke running       |  jh   #  ow   #   k   r   #  ah   #   n   #  ih  ng   #
Job reading        |  jh   #  ow   #   b   r  eh   #   d   #   #  ih  ng   #
Joke routing       |  jh   #  ow   #   k   r   #  aw   t   #   #  ih  ng   #
Jokes running      |  jh   #  ow   k   s   r   #  ah   #   n   #  ih  ng   #
Joke reading       |  jh   #  ow   #   k   r  eh   #   d   #   #  ih  ng   #
Jokes routing      |  jh   #  ow   k   s   r   #  aw   t   #   #  ih  ng   #
Joe brown ring     |  jh   #  ow   #   b   r   #  aw   #   n   r  ih  ng   #
---------------    |  --- --- --- --- --- --- --- --- --- --- --- --- ---
Frame              |  138 143 148 154 160 166 178 182 185 188 191 194 199 207
```

```
SF PHN   EF  AS_Score
-- ---   --  ---------
 4  JH   13  -1310710
14   Y   18  -3651243
19  OW   30  -4721217
31   B   36  -7351706
37   R   47  -8740488
48  AW   58  -10986376
59   N   63  -13269350
64  IH   69  -14345927
70  NG   80  -15624685
             -17695072
```

```
WORD (SFrm Efrm)  AS_Score
            <s> (  0    8) -1854350
       DH_IH_S  (  9   30) -3636935
          IH_Z  ( 31   43) -2076398
          ER_T  ( 44   58) -2327542
          T_AA  ( 59   75) -2777406
          AY_M  ( 76   90) -2103141
          AO_L  ( 91   98) -1334920
            OY  ( 99  108) -1511690
        IH_N_F  (109  127) -2990061
          ER_D  (128  136) -1316290
JOE_BROWNING(1) (137  210) -10917554
           </s> (211  256) -8499298
```

```
HW-alts
 1. JB
 2. j-B
 3. TB
 4. JOB
 5. JIB
 6. TAB
 7. JAB
 8. Jos
 9, JP
10. J-B
```

Phone Ensemble Outputs (PEO)
Speech-lattice (S-latts) word extractions
WPSR temporal bounds
Phone matrix input to dynamic bigram modeler
Second pass phone recognition
Integration Decision Metric
"Joe Browning" = JH OW B R AW N IH NG
"Joe Browning" is recognized by the Word/Phrase-Spotter
HW-alts
JB
2 S-trans
Speechalyzer transcript
3 S-latts
Speechalyzer lattice
1 HW-alt
HandWriting hypotheses
5 PEO
Phone Hypotheses
Multimodal Inputs
Word/Phrase hypotheses
4 WPSR-trans
Sketch/Chart Recognition
6
Joe Browning "JH Y OW B R AW N IH NG"
4
JB
Arrive
Joe Browning

Figure 6.14: To produce a refined pronunciation speech lattice term sequences spanning the WPSR temporal boundaries are aligned with the phone ensemble outputs. The resulting refined pronunciation becomes the *integration decision metric*, which serves to confirm that the WPSR recognition is present in the speech. This check allows the spelling and pronunciation of the proper name to be semantically attached to the handwritten abbreviation *JB*.

pronunciations, along the lines of what Roy [150] or Yu and Ballard [175] have outlined in their works), but for now we just expand the number of alternative pronunciations.

6.5 SUMMARY

This chapter has detailed the process of using the refined pronunciation produced by second-pass phone recognition as an *integration decision metric.* The role of an *integration decision metric* is to choose the appropriate group of inputs from which the final spelling and pronunciation should be taken, and then function as metric against which the final combinations can be scored. In-depth examples showed unimodal recognition failed in the recognition of a series of Gantt chart labels, whereas integration of redundant handwriting and speech succeeded in establishing correct understanding of the same labels — including learning dynamically the meaning of a new handwritten abbreviation.

Chapter 7

SHACER: Testing

This chapter examines various tests to which SHACER has been submitted — baseline recognition tests, tests of the validity of some of its primary assumptions, and tests of the efficacy of its approach to dynamic learning of out-of-vocabulary terms including abbreviations. These tests make it evident that SHACER's leveraging of multimodal redundancy improves recognition.

7.1 WORKING HYPOTHESIS

As it is evident that multimodal redundancy requires more energy than unimodal communication, there must be important communicative purposes driving its use (see Section 2.2.2). We believe that establishing a common ground of meaning is that purpose, and that people use redundancy as a conversational strategy to bolster their communicative effectiveness by drawing attention to the meanings of dialogue critical terms, as is supported by the evidence offered in Section 2.2. In the remainder of this chapter we will show how SHACER leverages its perception of these natural attention focusing events to significantly improve its computational ability to better understand a human-human interaction.

7.2 MULTIMODAL TURN END-POINTING

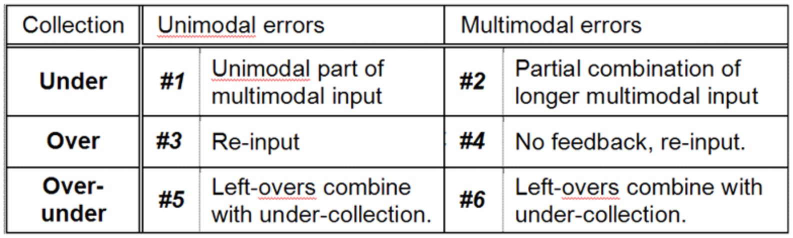
Segmentation of speech and ink was addressed in Chapter 5. As a multimodal system, SHACER also needs to segment different modal streams into turns of input. In a multimodal system predicting the end of user input turns can be complex. User interactions vary across a spectrum from single, unimodal inputs to multimodal combinations delivered either simultaneously or sequentially. Thus it is difficult for a multimodal system to know how to group and segment user input turns. Making incorrect turn segmentation decisions can adversely affect recognition, as the test results of SHACER's turn segmentation approach at the end of the section make clear.

7.2.1 Fixed-Threshold Wait-Times for Turn Segmentation

The current approach to the multimodal turn segmentation problem, as discussed in the literature, is to wait for some fixed threshold of time before assuming the end of user turn-input [136]. Recent research has sought to reduce this fixed wait time by the use of corpus-based, probabilistic [65] or user-adaptive models [73]) of input styles. The motivation for modeling this temporal threshold is to avoid under and over collection errors. Gupta *et al.* describes under-collection errors as having occurred when some user turn-inputs arrive after turn processing has already started. These types of errors are listed in Table 7.1 as numbers #1 and #2. On the other hand, over-collection errors are those in which users re-enter inputs due to a perception of system unresponsiveness. These are listed in Table 7.1 as numbers #3 and #4 [65]. As Johnston *et al.* [79] have pointed out, avoiding these types of errors is important when mistakes have disruptive or confusing side-effects. An example of such confusing effects would be under-collecting a multimodal pan command, which is composed of (a) drawing an arrow plus (b) saying "pan," by triggering a unimodal zoom command as soon as the arrow is drawn and disregarding whether any speech occurred or not.

Table 7.1 categorizes the types of temporal mis-combination errors that can occur in a multimodal system. Aside from under and over collection errors, we have added a third category, not described by Gupta *et al.*, that we term over-under collections. These occur when left-over inputs from previous commands remain available and combine with subsequent under-collections.

Table 7.1: Categorization of multimodal temporal turn-segmentation mis-combination errors by collection type and modality level.



Collection	Unimodal errors		Multimodal errors	
Under	***#1***	Unimodal part of multimodal input	***#2***	Partial combination of longer multimodal input
Over	***#3***	Re-input	***#4***	No feedback, re-input.
Over-under	***#5***	Left-overs combine with under-collection.	***#6***	Left-overs combine with under-collection.

Both Gupta *et al's* [65] and Huang *et al's* [73] recent studies assume the use of multimodal command interfaces, which alternate between accepting turns of user input and displaying the interpreted output. They both focus on minimizing under/over collection errors by better predicting how long to wait for the end of user turn input, using Bayesian modeling techniques. However, focusing solely on turn segmentation prediction does not adequately consider the underlying parsing mechanisms at work in a multimodal system.

7.2.2 Parsing and Edge-Splitting

This section explains our edge-splitting modification to the basic multimodal chart parsing algorithm. Without this modification many integrations of speech and handwriting information would be systematically dis-allowed. Therefore it is important to understand how edge-splitting works.

In Section 4.3.2 we reviewed the basic temporal chart parsing algorithm that underlies SHACER's use within Charter. For convenience we give the formulas again here. The $*$ in Equation 7.1 is an operator that combines two constituents according to the rules of the grammar. Constituents are designated as terminal sequences from vertex to vertex, and both the vertices and constituents are linearly ordered.

$$Chart(i,j) = \bigcup chart(i,k) * chart(k,j) \tag{7.1}$$

As Johnston points out [79], in a multimodal context linearity is not assured, because input from different modal constituents can well be temporally overlapped. Thus he defines the basic temporal, multimodal chart parsing algorithm as:

$$\begin{aligned} multichart(X) &= \bigcup multichart(Y) * multichart(Z) \\ where X &= Y \bigcup Z, Y \bigcap Z \neq \varnothing, Y \neg \varnothing, Z \neg \varnothing \end{aligned} \tag{7.2}$$

Constituent edges in a multimodal parse space cannot be identified by linear spans. This is the meaning of Equation 7.3. Instead they are identified by unique sets of identifiers (e.g. multichart([s,1,1,0],[g,2,2,1])), each of which specify the constituent's mode of origin, recognition sequence number, position on the list of alternate recognitions, and semantic interpretation. This identification axiom maintains the critical constraint enforced by linearity that a given piece of input can only be used once in a single parse. Commands with intersecting IDs are different interpretations of the same input, and are thus ruled out by the non-intersection constraint in equation (2) above. This means that there can only be one correct interpretation acted upon for each set of inputs. Therefore once that best scoring command is chosen and executed all constituent edges from that command are removed from the chart.

This removal policy means that all constituent edges, which participate in an otherwise correct interpretation of partial, under-collected input, are then no longer available to participate in a subsequent interpretation of fully collected turn-inputs, because their IDs would intersect. This is the underlying issue in multimodal parsing that makes under-collection a general problem.

$$multichart([id, 2, 1, 0]) \Longrightarrow multichart([mmid, 2, 1, 0]) \tag{7.3}$$

SHACER's solution is to (1) filter all messages of an appropriate type (e.g. all ink-gestural edges), (2) clone them — changing only the input mode symbol identifier (**id** $\Longrightarrow$ **mmid**, Eq. 3), and (3) put the cloned edges back on the chart. It then enforces the constraint that edges with the new input mode symbol identifier (e.g. **mmid**) only participate in subsequent multimodal interpretations. They can no longer be interpreted unimodally. These split-edge clones are periodically removed from the chart just as other edges are removed, based on an edge-defined time-out period. To allow for long distance associations across modes, edge time-outs are ignored until at least the next edge of the same type arrives on the chart. Thus edge-splitting, in conjunction with an under-specified display [88], solves the underlying problem of under-collected, unimodal interpretations starving subsequent multimodal interpretations by removing the edges needed for multimodal integration.

7.2.3 Testing Edge-Splitting in Charter

Figure 7.1 depicts the use of edge-splitting in our Charter Suite prototype application for ambient-cumulative multimodal recognition of a multiparty scheduling meeting. The upper half of the diagram in Figure 7.1 shows an example task-line, labeled *office*, and diamond-shaped milestones marking the temporal availability of office space (abbreviated as *Avail*). The bottom half of the diagram shows an example task-line, labeled *Buy Computer*.

Without edge-splitting, ink-gestures that temporally precede the spoken utterances with which they are associated fire unimodally producing incorrect interpretations: (middle column of Figure 7.1), *trail* (for *avail*), *lay computer* (for *Buy Computer*). These were under-collection errors. Their respective edges were removed from the chart disabling subsequent multimodal recognition. Also, for abbreviation interpretations based solely on

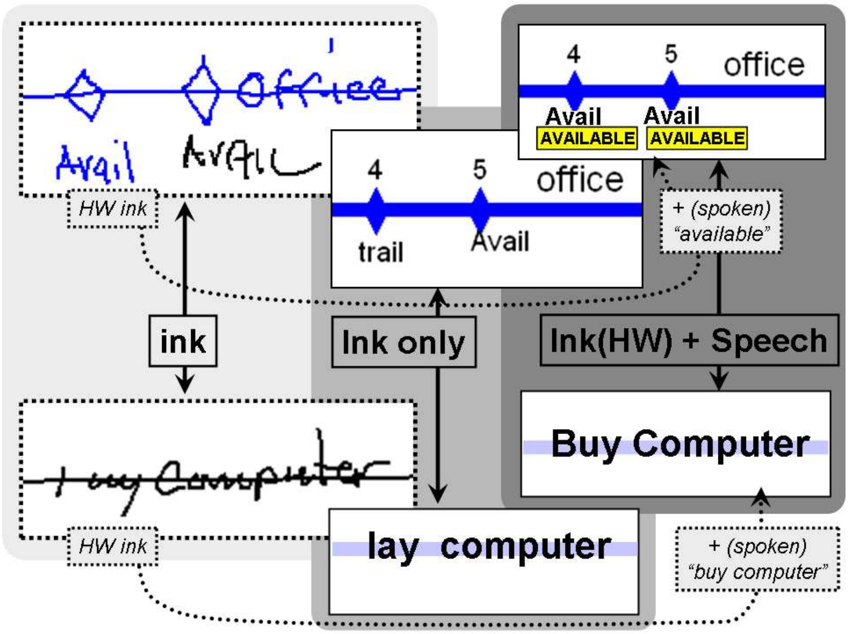


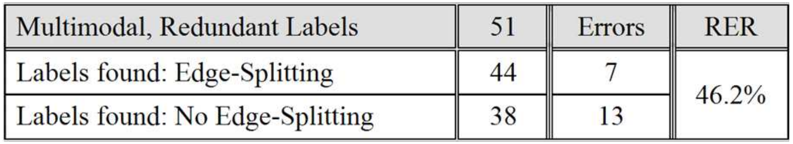
Figure 7.1: **Left-column:** Gantt chart ink. **Middle column:** Edge-splitting disabled, so interpretations are based on Type 1 under-collection errors (see. Table 7.1). **Right column:** Edge-splitting enabled, so errors have been corrected by subsequent multimodal interpretations. Lighter colored ink (upper left) illustrates no-wait, under-specified recognition display.

under-collected ink input (middle column of Figure 7.1: *trail*, *Avail*), there were no semantic glosses (e.g. Figure 7.1, upper right, gray text boxes containing "AVAILABLE"). These glosses were produced only by integration with speech, via SHACER. Without edge-splitting under-collection errors disable the multimodal integration necessary for semantic glosses.

With edge-splitting enabled the incorrectly interpreted unimodal ink-gesture edges were split, and their multimodal clones put back on the chart. These split edges then combined with their respective spoken utterances producing correct multimodal interpretations (shown in the right column of Table 7.1), which replaced the incorrect unimodal interpretations.

Table 7.2 shows the test results for using edge-splitting. These tests results are from processing the *G* series of meetings. There were 51 constituent labels in the five finished Gantt Charts created during this meeting series. All 51 were presented multimodally, i.e. handwritten and spoken redundantly. Without edge-splitting there were 13 multimodal label recognition errors. With edge-splitting there were only 7. Counts were determined by

Table 7.2: Test Results for multimodal constituent labeling, with Relative Error-rate Reduction (RER) for Edge-Splitting.



Multimodal, Redundant Labels	51	Errors	RER
Labels found: Edge-Splitting	44	7	46.2%
Labels found: No Edge-Splitting	38	13	

visual inspection of system output (e.g. Figure 7.1, middle and right columns.). Therefore edge-splitting yielded a relative error rate reduction (RER) of 46.2% — significant by a McNemar test (p = 0.03).

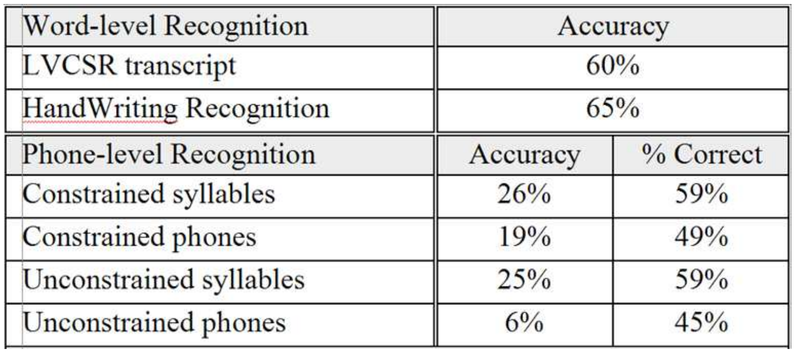
We note that in two of the six error instances corrected by SHACER's edge-splitting technique speech was associated with a preceding ink-gesture that was between 3-22 unrelated utterances and 45-250 seconds earlier. Thus, edge-splitting mitigated under-collection errors for both temporally distinguished turns (Figure 7.1, *buy computer*), and for input groupings that were structurally distinguished based on their redundancy relations despite long temporal/turn distances (Figure 7.1, *Avail*). Turn segmentation based solely on temporal thresholds (e.g. adaptive temporal threshold prediction [65, 73]) could not address such integration errors across long-distance, structurally-distinguished groupings.

In our test set of five meetings there were 500 utterances and 183 gestural/ink inputs. Most chart constituents could only be recognized unimodally (e.g. axes, tickmarks, tasklines, milestones), while others (e.g. handwritten labels) could be interpreted either unimodally or multimodally. All gestural inputs were initially interpreted unimodally. In edge-splitting mode, roughly 24% (44/183) of gestural inputs were also interpreted multimodally (Table 7.2), whereas in non-edge-splitting mode 21% (38/183) were interpreted multimodally. In non-edge-splitting mode the unimodal recognition rate was roughly 87%, while in edge-splitting mode it was 89%. Thus, in our test series, with competing unimodal/multimodal recognitions, edge-splitting never confounded or degraded recognition, it only significantly improved recognition.

7.2.4 Baseline Word and Phone Recognition

The counts of individual utterances and individual ink-gestures vary depending on the settings used to parameterize Speechalyzer's endpointing mechanism and the version of Charter that is used for performing gesture segmentation. For example, across SHACER's development test set of five meetings (*G1-G5*) for the tests reported in the rest of this chapter there are a total of 402 individual utterances and 181 individual ink-gestures. These counts differ from those given in Section 7.2.3 for the reasons stated above.

Table 7.3: Recognition accuracy for word, phone, and handwriting recognizers over SHACER's five meeting development test set (*G1-G5*, 360 speech and 51 handwriting instances). Accuracy is (Correct − (Substitutions + Insertions + Deletions))/Total, while Percent (%) Correct is (Correct/Total).



Word-level Recognition	Accuracy	
LVCSR transcript	60%	
HandWriting Recognition	65%	
Phone-level Recognition	Accuracy	% Correct
Constrained syllables	26%	59%
Constrained phones	19%	49%
Unconstrained syllables	25%	59%
Unconstrained phones	6%	45%

The LVCSR transcript accuracy, handwriting recognition accuracy (MS Tablet PC Handwriting recognizer), and accuracy and correctness for each of the ensemble of phone recognizers are given in Table 7.3. The speech results are computed from only 360 of the total 402 utterances because many utterances are short filled-pauses and these are not scored. The handwriting results are computed from only the 51 handwriting events that occurred as part of the 181 total ink input events. We define *accuracy* and *correctness* by the standard NIST definitions for determining word error rates, as implemented in the CSLU Toolkit [49]. The actual formulas for *accuracy* and *correctness* are given in the caption of Figure 7.3. For determining phone error rates, phones are treated as words and compared on the basis of their spelling, using a standard Levenshtein edit distance, as is done in computing word-level error rates. This is the standard method of reporting phone error rates [57].

Speech Using constrained syllables as sub-word units yields more accurate phone recognition because phonetic sequences and word structure are better accounted for; however, constrained and un-constrained phone versions, although much less accurate, are still used because they provide finer phonetic boundary information and more directly reflect feature-level acoustic information important for our alignment mechanism.

7.2.5 Word/Phrase Spotter Out-Of-Vocabulary (OOV) Enrollment

SHACER uses a word/phrase-spotter as the target for OOV enrollment, as opposed to enrollment solely in the LVCSR dictionary and language model. The WPSR output indicates with high likelihood that a dialogue-critical term has been uttered. The number of enrolled terms per meeting is presently small (about 8-10), but as the system's capabilities grow we hope to learn whole phrases or sub-grammars, which can be used to complement the LVCSR output. Such a capability lends itself to techniques for contextually constrained language modeling as in [52, 62, 149].

SHACER's WPSR uses the same two-level RTN architecture used by the MNVR recognizer (Section 4.3.2), but the levels are switched. In MNVR the carrier-phrase word level is primary and the syllabic out-of-vocabulary level is secondary, while for WPSR the syllabic level is primary (to cover all non-enrolled terms at a phonetic level) and the enrolled word level is secondary. In both implementations the transition from primary to secondary grammars is gated by an empirically determined confidence threshold, so that in the carrier-phrase case the out-of-vocabulary hypotheses are high confidence and in the WPSR case the enrolled word recognitions are high confidence.

A further augmentation to WPSR was to allow the inclusion of virtual sub-grammars. These are illustrated by the <*place_holder_word_phrase_name*> virtual sub-grammar in Section B.4.1's [*word_phrase*] grammar. By virtual we mean that initially such a sub-grammar has no terminals, but only empty place-holders. There was a virtual sub-grammar for both taskline names and milestone names, so the semantics of the new word are used to determine which virtual grammar will receive the new vocabulary. Although the place-holder terminals cannot participate in recognition, they nonetheless have a specific location within the primary grammar at which they can occur (see Section 4.3.2). When enrolled words replace place-holders at the specific locations in which they are grammatically licensed, then subsequently the new words are licensed to occur in those grammatical locations.

7.3 TWO PHASE PROCESSING

Meeting processing is currently off-line, requiring on the order of 10-20 times real-time on a dual 3G+ workstation. Utterance transcript logs along with the raw ink recording are processed in a first phase pass over the multimodal data. This pass produces separate utterance files based on boundaries from the LVCSR transcript log, as well as logs of phone-level transcripts for three out of the four phone recognizers (the slave recognizers). The fourth phone recognizer, the master, must be re-run in the next phase so that phase two's dynamic second-pass speech processing can occur, which comprises the main processing steps of *refinement* and *integration*.

The second phase of processing uses utterance transcripts along with lattice logs and phase-one recognition message logs as input. This phase performs actual integration of handwriting and speech hypotheses. The output is a labeled Gantt chart. When learning mechanisms are being employed the knowledge from learning is stored in file-based accumulators (see Appendix C), which serve as updaters for (1) the word/phrase-spotting recognizer's dictionary, (2) the handwriting recognizer's dictionary, (3) the handwriting recognizer's reinforcement table, which biases recognition in favor of previous high-confidence, integrated recognitions, (4) an abbreviation table in the master phone recognizer, and (5) a prefix/suffix mis-recognition reinforcement table in the master recognizer that associates high confidence recognitions with their list of related mis-recognitions and significant affixes (so that when affix mis-recognitions are discovered, which are very highly correlated to a previous recognition, that consistent recognition error can be transformed into its correct interpretation). These accumulators are described in the next section.

7.4 FILE-BASED ACCUMULATORS FOR LEARNING PERSISTENCE

Examples of the four file-based accumulators, which are loaded into SHACER when processing the *G5* meeting in learning mode, are given in Appendix C, where they are described in detail. Each file in this appendix represents the actual learning accumulated over the development test set's first four meetings (e.g. *G1-G4*). Brief conceptual descriptions of what the knowledge stored in these files accomplishes is given below.

7.4.1 Adding to Speech and Handwriting Dictionaries

The file which accumulates additions to the systems vocabulary is shown in Section C.1. It is the basis for updating both the word/phrase-spotting recognizer's (WPSR's) and the handwriting recognizer's run-time dictionaries. At the start of meeting processing each message in the file is read in and processed. The information in each file entry specifies (1) the term being enrolled in the WPSR recognizer's dictionary along with its specified pronunciation (e.g. *Joe_Browning*, *'JH OW B R AW N IH NG'*), and (2) the individual words, *Joe* and *Browning*, being enrolled in the handwriting recognizer's dictionary. Words that are in the handwriting recognizer's dictionary tend to be better recognized than words that are not.

7.4.2 Expanding the Master Speech Recognizer's Abbreviation Table

Entries in the `shacer_vocab.txt` file (Section C.2) are used to populate the master phone recognizer's abbreviation expansion table. This table supports lookups of the handwriting recognizer's letter sequence hypotheses, so that when a match occurs between a handwriting recognizer alternative and a table entry the abbreviation expansion can be immediately recovered. This aids in identifying instances of redundant speech.

7.4.3 Biasing Handwriting Recognition

The file that holds weight-biasing entries is shown in Section C.4. Its entries are used to populate the handwriting recognizer's weight biasing mechanism, which boosts more frequently recognized terms to higher positions on the output alternates list. Each addition increases a term's frequency count. Terms with higher frequency counts have their recognition scores proportionally increased and thus can move up in the list of output alternatives — in effect becoming easier to recognize the more often they have been seen in the past.

7.4.4 Affix Tracking for Mis-Recognition Correction

```
(Example D)
    CB = [cos, as, cy, cd, cry, coy, ay, Coy, cis]
    CB = [iB, SB, Cts, EB, eB, cps, cB]
```

Finally, entries in the affix reinforcement file, shown in Section C.3, are used to populate the master phone recognizer's prefix/suffix mis-recognition reinforcement table. Example mis-recognitions of handwriting events are listed in **Example D**'s entry for the abbreviation, *CB* = *Cindy Black*. If such a mis-recognition occurs again for a subsequent handwriting recognition event, and it can be uniquely linked to one of these previously seen incorrect forms that is strong evidence for *CB* having actually been handwritten. For example, during meeting *G5*, as described below in Section 7.6.1, a handwritten label for *CB* is poorly recognized due to an ink skip over the letter *C*. This results in a recognition of *iB*. Given the table entries shown in **Example D** above, this mis-recognition is unique to *CB*, and it is therefore replaced by the correctly associated *CB*. Judgements about when a mis-recognized affix can be replaced are dependent on a measure of its uniqueness. Only uniquely mis-recognized affixes can be replaced.

7.5 PLAUSIBILITY TESTING

7.5.1 Speech/Handwriting Combination Benefits

SHACER's approach to producing the refined pronunciation of redundantly presented terms, is to align phone ensemble outputs with LTS transformations of associated handwriting, and then use a phone sequence model extracted from that alignment to constrain second-pass phone recognition. How plausible is this approach? How potentially effective could it be? Does including information from the handwriting LTS phone sequences in the positional phone bigram model actually help second-pass recognition or hurt it?

To illustrate the general plausibility and potential benefit of combining speech and handwriting information in a constrained second pass speech recognition, and to get some idea of the upper bound of improvement in terms of phone accuracy that is achievable by combining handwriting information with phone ensemble recognitions we performed an experiment using the correct phone transcript of each of the 360 development utterances as if it were the letter-to-sound transformation of the handwriting. Thus it was as if we had perfectly recognized and transformed redundant handwriting for every word spoken in every utterance, and this information was combined with the speech information via our positional phone-bigram model.

Figure 7.2 shows the two test conditions. In the first condition, *speech+speech*, only speech information is used. Because the multiparser rules on which integration rests expect inputs from both speech and handwriting, copying the speech information and using it as if it were handwriting information means that a parallel rule system does not have to be written. This saves time, and using the copied speech in this way still adds no new information to the methods that extract phone sequence bigrams. Thus the repeated speech information does not change the outcome of the second pass phone recognition. The second condition, *speech+pseudo-HW*, simulates the addition of perfect handwriting. Any change in accuracy of second-pass refined pronunciations using the *speech+pseudo-HW* compared to the baseline *speech+speech* can demonstrate how much adding this type of phonetic information can possibly help, if at all.

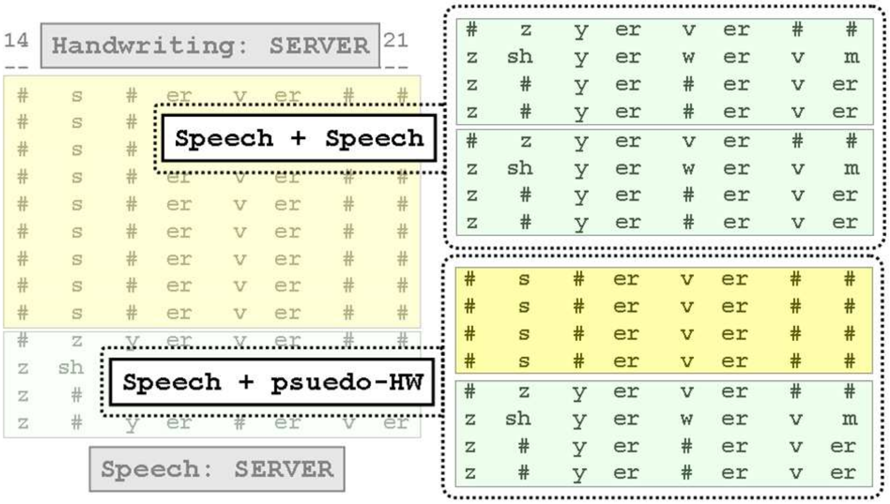


Figure 7.2: Test conditions for examining the plausibility and efficacy of combining phonetic information from handwriting with speech — in the *speech+pseudo-HW* condition. The control condition is *speech+speech* in which no new information is added.

Table 7.4: Phone-level recognition accuracy over SHACER's five meeting development test set (*G1-G5*, 360 speech utterances). Accuracy is (Correct − (Substitutions + Insertions + Deletions))/Total, while Percent (%) Correct is (Correct/Total). Key: *pb* = positional phone-bigram information used, *no-pb* = no positional phone-bigram information used.

Phone-level Recognition	Accuracy	% Correct
pb - (speech+speech)	24%	42%
pb - (speech+pseudo-HW)	48%	54%
no-pb - (speech+speech)	8%	41%
no-pb - (speech+pseudo-HW)	15%	41%

Results are shown in Table 7.4, and in the charts depicted in Figure 7.3. The leftmost of the charts in Figure 7.3 broadly compares the control *speech+speech* condition to the active *speech+pseudo-HW* condition. It clearly illustrates that combining information from handwriting with speech can improve the phonetic accuracy of second pass phone recognition, irrespective of whether positional phone bigram modeling is used or not. The vertical axis in each of Figure 7.3's charts is the percent accuracy.

The middle and right charts in Figure 7.3 show that there were two dimensions of change. The first dimension of change was with respect to the use of positional phone-bigram information (*pb*) to constrain second-pass speech recognition. The right-most chart in Figure 7.3 shows that there were two states: (1) *pb* (indicating that positional phone-bigram info was used, Table 7.4, rows 1-2), and (2) *no-pb* (indicating that no positional phone-bigram info was used because every bigram likelihood was set to 1.0, Table 7.4, rows 3-4).

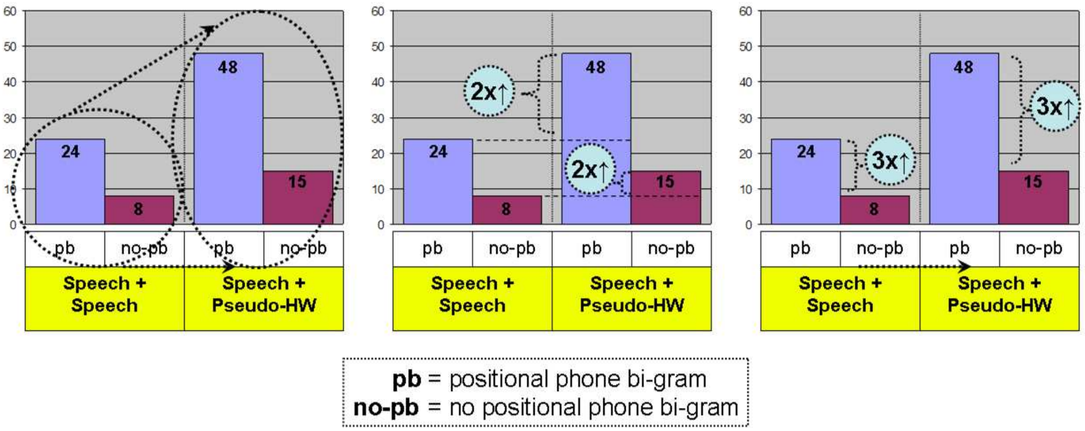


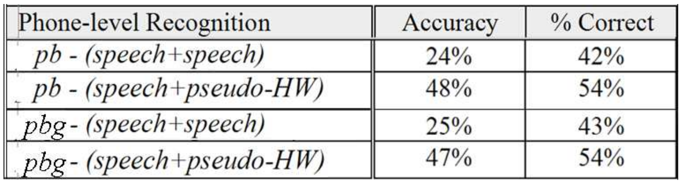
Figure 7.3: Charted results (from Table 7.4) for testing the efficacy of combining handwriting with speech information to improve phonetic recognition accuracy (as measured on the vertical axes).

Across the first dimension of *pb/no-pb* there was a tripling of accuracy from 8% *no-pb* to 24% *pb* in the *speech+speech* condition and from 15% *no-pb* to 48% *pb* in the *speech+pseudo-HW* condition. Thus using the positional phone-bigram model's statistical phone-sequence information was very important.

The second dimension of change, which is depicted in Figure 7.3's middle chart, was what type of pseudo-handwriting information was used: either (a) a copy of the utterance's existing phone ensemble output sequences, or (b) the correct phone transcript for the utterance. State (a) represents the *speech+speech* condition, meaning that no new phone sequence information was added because the pseudo-handwriting was identical to the existing speech information. These results are shown in Table 7.4's rows 2 & 4. State (b) represents the *speech+psuedo-HW* condition, meaning knowledge of what the perfectly recognized handwriting's letter-to-sound phones would be was added in. Results for this state are shown in Table 7.4's rows 2 & 4).

Across the second dimension of pseudo-handwriting conditions (*speech+speech* compared to *speech+pseudo-HW*) in Figure 7.3's middle chart there was a doubling of accuracy, from 8% to 15% in *no-pb* mode and from 24% to 48% in *pb* mode. This tells us that combining phone sequence information from handwriting with phone sequence information from speech can indeed have a large impact on the accuracy of second-pass speech phone recognition. Findings across both dimensions of change strongly validate our approach of integrating letter-to-sound transformations from the handwriting with outputs from our ensemble of phone recognizers to produce refined phone sequence hypotheses.

Table 7.5: Phone-level recognition accuracy over SHACER's five meeting development test set (*G1-G5*, 360 speech utterances). Accuracy is (Correct − (Substitutions + Insertions + Deletions))/Total, while Percent (%) Correct is (Correct/Total). Key: *pb* = positional phone-bigram, *pbg* = phone-bigram (i.e., no positional information used).



Phone-level Recognition	Accuracy	% Correct
pb - (speech+speech)	24%	42%
pb - (speech+pseudo-HW)	48%	54%
pbg - (speech+speech)	25%	43%
pbg - (speech+pseudo-HW)	47%	54%

Also, as shown Table 7.5, not using positional information as part of the phone bigrams (i.e. the *pbg* condition) only marginally improves accuracy in the *speech+speech* condition (from 24% to 25%), while actually degrading accuracy in the *speech+pseudo-HW* condition (from 48% to 47%). So we cannot yet show statistically that positional information helps in contributing to the overall highly beneficial effect of using phone bigram constraints during second pass phone recognition.

To summarize, in Table 7.4 and Table 7.5 we compared three methods of constraining second-pass phone recognition. The constraints are three different phonetic sequence models:

1. Use of positional phone bigram models (*pb* condition in Table 7.4 and Table 7.5).

2. No use of positional phone bigram models (*no-pb* condition in Table 7.4), which is done by setting all bigram scores to 1.0 so that sequence information doesn't matter during second-pass decoding.

3. Use of phone bigram models that have no positional information (*pbg* condition in Table 7.5).

The *pb* condition compared to the *no-pb* condition helped greatly, tripling accuracy rates. However, there was no clear advantage to either using positional information, *pb* condition, or not using it, the *pbg* condition.

7.5.2 Testing Articulatory-Feature Based Phonetic Distances

Given a phonetic representation for each hypothesis in the list of handwriting recognitions (via a letter-to-sound transformation) SHACER needs to discover what segments of speech are associated with it. Standard edit-distance [103], based on the spelling of phone names, does not provide an accurate distance measure for this purpose. So SHACER uses an approach put forward by Kondrak [96] for phonetic articulatory-feature based alignment (see Section 5.3.3).

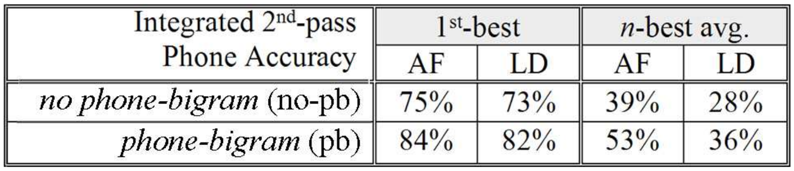
Articulatory Feature-based Alignment

To test the effectiveness of articulatory-feature based alignment we ran our proof-of-concept system over the five meetings of the development test set and compared the accuracy of second-pass speech phone sequences for 14 OOV terms using a system-wide standard Levenshtein edit distance (LD) versus using an articulatory-feature based distance metric (AF). Thus, this experiment represents two runs of the SHACER system on the G series of meetings, using both handwriting and speech information. In one run the alignment distance metric was standard Levenshtein edit distance (LD run), and in the other run the articulatory-feature based alignment metric was used (AF run). Both runs used the full integrative mechanisms of the system with both handwriting and speech input. The purpose was to test which alignment metric worked better. This test was run on OOV terms (although there were only a small number available) because for terms that are in-vocabulary it is futile to try to improve their pronunciations — they already have the canonical pronunciations which are the standard against which improvement would be measured.

It would have been desirable to have more than 14 OOV terms to test on. However, since these were full system tests across the entire G series of meetings it was not possible to create more OOV test terms by removing vocabulary from the transcribing recognizer's dictionary and language model. Any such removal would necessitate rebuilding the language model of the transcribing recognizer (CMU Sphinx 3.5), and the resources were not available to do that. There were other meeting series from the same SRI year 2 meeting corpus, but each other series had some technical collection issue, like poor microphone quality or non-standard chart diagrams, that disqualified it.

The results of testing the effectiveness of an articulatory-feature (AF) based distance measure versus a standard Levenshtein edit distance (LD) based measure, are shown in Table 7.6. Table 7.6's 1st-best columns shows an absolute 2% gain in accuracy due to

Table 7.6: The effect of system-wide articulatory-feature based alignment (AF) versus standard Levenshtein distance alignment (LD) on pronunciation accuracy of second-pass phone-level interpretations over a set of fourteen OOV terms. Key: pb = positional phone-bigram; no-pb = no positional phone-bigram.



Integrated 2nd-pass Phone Accuracy	1st-best		*n*-best avg.	
	AF	LD	AF	LD
no phone-bigram (no-pb)	75%	73%	39%	28%
phone-bigram (pb)	84%	82%	53%	36%

the use of the articulatory-feature based (AF) metric (from 73% to 75% and from 82% to 84%). The benefit of using the AF versus LD metric is better reflected in Table 7.6's n-best average columns. This is because the 1st-best alternative often reflects the influence of LVCSR word-lattice or word/phrase-spotter outputs. These are more salient combination indicators than full handwriting/phone-ensemble alignments (see Section 6.1.1). Table 7.6's n-best list columns are more reflective of actual second-pass recognition alternatives. For these alternatives, which depend directly on the alignment of handwriting and phone-ensemble phonetic sequences, we see that even without using constraints from the positional phone-bigram model (*no-pb* condition) there is a 39% relative increase in average accuracy (from 28% to 39%), and when the positional phone-bigram model is used there is a 47% relative increase in average accuracy (from 36% to 53%). This again reflects the benefit of using dynamic positional phone-bigram modeling to constrain second-pass phone recognition. Compared to not using it (*no-pb*) there is a 36% relative reduction in error rate (AF/*no-pb* 25% to AF/*pb* of 16%).

These results indicate that using an articulatory-feature based distance metric (AF condition) is better than using only Levenshtein edit distance (LD condition). However, because these statistics are figured on the processing results from only 14 OOV instances, they are indicative but not yet significant. To see the difference that these effects make we must turn to a closer examination of the overall system output.

Articulatory-Feature Based Metric for Learning the Meaning of Abbreviations

When a user at the whiteboard handwrites a label and says it, that instance of multimodal redundancy triggers learning and enrollment into the Word/Phrase-Spotting Recognizer (WPSR), the handwriting recognizer, and other reinforcement tables (see Section 7.4). For example, saying, "Joe Browning," while writing it out in full triggers learning and

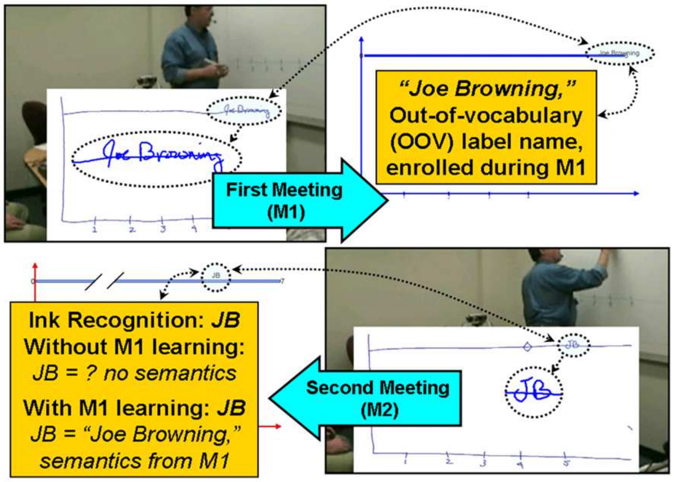


Figure 7.4: Introducing a new term via multimodal redundancy: (*G1*) writing the name of a new hire, *Joe Browning*, to label a schedule taskline on a Gantt chart while saying, "This line represents Joe Browning." (*G2*) Referring to *Joe Browning* again, while handwriting an abbreviation, *JB*, for his name. *G1* learning provides the meaning of the *JB* abbreviation in *G2*.

enrollment (Figure 7.4). Once the WPSR's dictionary contains *Joe Browning*, then when next spoken it is recognized by WPSR (hopefully). If it is also then handwritten as an abbreviation a series of checks and comparisons will be triggered as described in Section 6.4.2..

Table 7.7: Results for learning abbreviation semantics over the 5-meeting development test set, using system-wide articulatory-feature based alignment (*AF*) versus standard Levenshtein distance (*LD*) alignment. Key: pb = positional phone-bigram constraint used, no-pb = no positional phone-bigram constraint used. ER = error rate.

	Labels (letter-correct/semantically correct)						ER
Meeting Number	1	2	3	4	5	TOTAL	
Total Possible	0/0	1/1	8/8	6/6	9/9	24/24	0%
AF	0/0	1/1	8/6	4/4	3/3	16/14	42%
LD	0/0	1/1	8/4	4/3	3/3	16/11	54%

If there is an abbreviation match discovered, then any new pronunciations for "Joe Browning" are added to (1) the WPSR dictionary, and the abbreviation, *JB*, is added to (2) the handwriting recognizer's dictionary and (3) reinforcement table, as well as to (4) the table of abbreviations and their expansions in the master phone recognizer, and also is processed for significant prefixes or suffixes which are added to (5) the prefix/suffix mis-recognition reinforcement table (Section 7.4). Information in these five tables is what persists and accumulates across meetings when the system is in learning mode. Thus if *JB* has been recognized and associated with *Joe Browning* in G2, because *Joe Browning* was fully enrolled in G1 (Figure 7.4), then subsequently *JB* is more readily recognized, and the system knows how to associate it with *Joe Browning* — that is, the system knows that the local meaning of *JB* (i.e., its semantics) is the expanded string, *Joe Browning.*

If we examine the system-wide effects of articulatory-feature based alignment versus Levenshtein distance based alignment on the learning of abbreviation semantics we find a 22% relative reduction in error rate, as shown in Table 7.7 (54% error rate for LD/*pb* to 42% error rate for AF/*no-pb*). Because of the small sample size this is a marked but not significant reduction.

7.6 TEST OUTCOMES

7.6.1 Results on Development Test Set

Before presenting test results, an example section of Gantt chart label recognitions will be explained. This example illustrates the difficulties involved and provides some context for explaining the test outcomes.

Figure 7.5 illustrates a portion of the sketched Gantt chart created during the fifth meeting of the development test set. Each meeting in the series was about various aspects of hiring three new employees (e.g. *Joe Browning*, *Cindy Black* and *Fred Green*). None of these two-word names were in either the dictionary or language model of the LVCSR, so they were not well recognized. This is reflected in the utterance samples in Figure 7.5 where “Sunday Black” is recognized for *Cindy Black* and “[In]fer agreeing” for *Fred Green.* The ink-only recognitions are shown in the *No Learning* box in the upper-right of the diagram.

This example (Figure 7.5, left-side) shows a task-line and diamond-shaped milestone. The task-line is for the arrival of the new hires and is labeled with the first-letter abbreviation for each of them (*JB*, *CB*, *FG*). None of the names were correctly recognized by the LVCSR; but, all were learned in previous meetings, so using the techniques described in Section 6.4.2 their semantics were assigned correctly in four of the five instances, as shown in Figure 7.5's *Learning* box in the lower right. This was accomplished because *Joe Browning* and *Fred Green* were recognized by the word/phrase spotter. However, *Cindy Black* was not. Rather, *CB* had been previously learned as an abbreviation, and even though a significant ink skip caused the *C* in *CB* to be missed almost entirely (in this example the ink recognition is *iB*), the correct interpretation was still recoverable. It was recoverable because one of the skipped ink's interpretations exactly matched a previously seen mis-recognition suffix associated uniquely with *CB* (see Section 7.4.4).

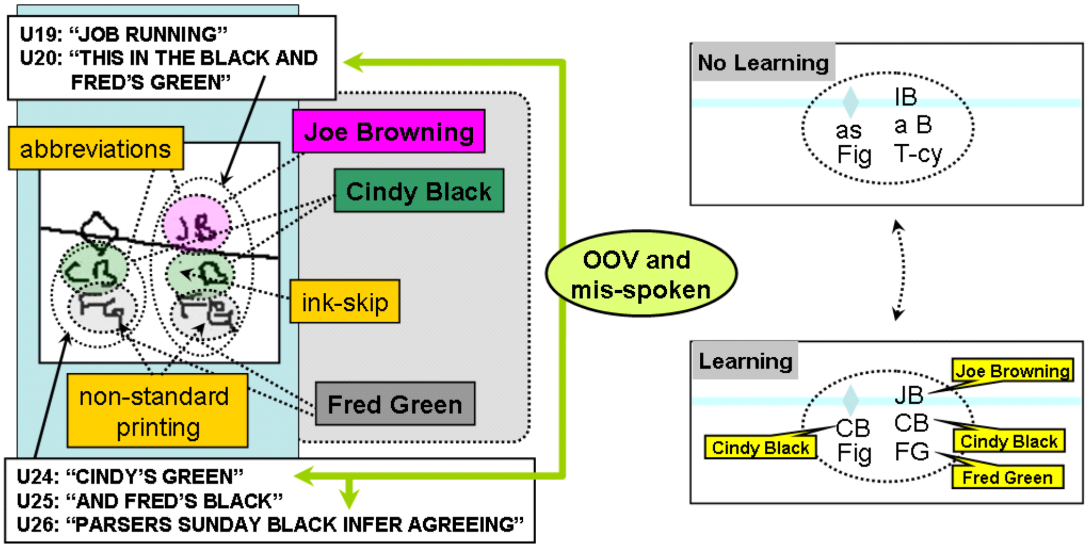


Figure 7.5: A portion of the Gantt chart created in the final development set meeting: abbreviations, ink skips, non-standard printing, and OOV or mis-spoken utterances made recognition and understanding challenging.

The arrival time of *Cindy Black* and *Fred Green* (*CB* and *FG*) was temporally marked by a diamond-shaped milestone. While handwriting their abbreviations the speaker mistakenly said, "Cindy Green," and, "Fred Black," mixing up their names. Later in utterance 26 (Figure 7.5, U26) he corrected himself, saying, "I'm sorry, Cindy Black and Fred Green," which again was poorly recognized. Thus, neither of these two abbreviations could be identified from speech evidence with integrative methods alone. *CB*'s semantics was recovered based on previous learning of the hand-written abbreviation and thus its known association to *Cindy Black*, but this did not work for *FG* because the non-standard *G* confounded the generation of any previously learned interpretations or mis-interpretations.

Ink-skips and mis-spoken names — these are some of the problems in recognition and integration that SHACER must deal with. Across all five meetings of the development test set there were 51 scorable labeling events (Table 7.8): 27 unabbreviated terms, and 24 abbreviations. Table 4 shows results in non-learning and learning conditions, with unabbreviated label results grouped to the left of abbreviated label results. They are grouped separately because the format of abbreviation scores is *x/y*, where *x* is the number of letter correct (i.e. lexically correct) labels and *y* is the number of semantically correct labels. Note that an abbreviated label was only considered correct when it was both lexically and semantically correct. In that case a hover label with its associated expansion was also shown in the Charter display (see examples in the Figure 7.5 lower right *Learning* area). In non-learning mode abbreviations were never semantically correct, because the system had no way of making the associations. A letter-correct label means that it is spelled correctly. A semantically correct label means that if the label is an abbreviation, like CB, then it is accompanied by a semantic hover-label with its correct abbreviation expansion, which for CB is Cindy Black.

Learning in these SHACER tests has a twofold definition:

1. The ability to make inferential associations across inputs modes, which add or confirm integrated information not available in either mode alone.

2. The ability to remember integrated information both during a single meeting and across a series of meetings.

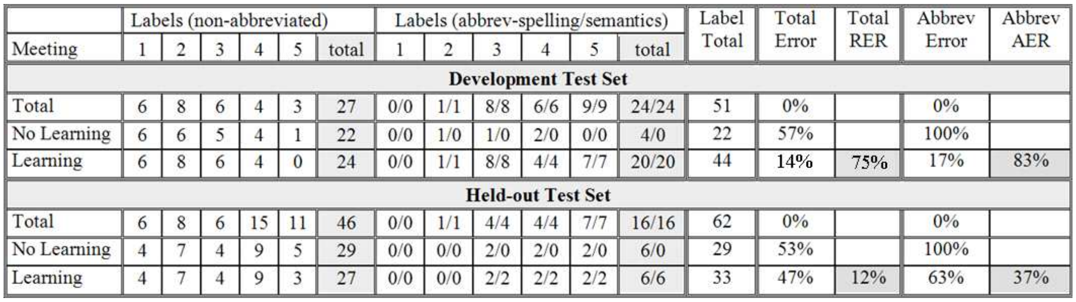
Thus, in non-learning mode, recognition of chart labels devolves to sketch/handwriting recognition, because no inferential associations are made and there is no ability to remember a previous recognition or use it to bias future recognitions.

Across the development test series (*G* series), which was used for development of the system, the use of learning resulted in a 75% relative reduction in error rate for label recognition generally (significant by McNemar test, p<=4.77e-07). This result is shown in Table 7.8's *Total RER* column, where *RER* = *Relative Error-rate Reduction*. For abbreviated labels, learning accounted for an 83% absolute reduction in label recognition error rate (significant by McNemar test, p<=1.91e-06). This result is shown in Table 7.8's *Abbrev AER* column, where *AER* = *Absolute Error-rate Reduction*.

7.6.2 Results on Held-Out Test Set

The held out test set also was a series of five meetings (*H1-H5*), with a scenario similar to that of the development test set, and with the same meeting participants. However, the held-out meeting set had some substantial differences; for example, there were 70% more non-abbreviated labels, and 50% fewer abbreviations. The participants in the meeting corpus were free to vary their meetings within the guidelines of the scripted instructions. This arbitrary variation accounts for these differences in number of abbreviations used.

Table 7.8: Summary results on development test set and held-out test set. Abbreviated labels are considered correctly recognized only when both spelling and semantics are correct. RER = Relative Error rate Reduction. AER = Absolute Error rate Reduction.



	Labels (non-abbreviated)						Labels (abbrev-spelling/semantics)						Label Total	Total Error	Total RER	Abbrev Error	Abbrev AER
Meeting	1	2	3	4	5	total	1	2	3	4	5	total					
Development Test Set																	
Total	6	8	6	4	3	27	0/0	1/1	8/8	6/6	9/9	24/24	51	0%		0%	
No Learning	6	6	5	4	1	22	0/0	1/0	1/0	2/0	0/0	4/0	22	57%		100%	
Learning	6	8	6	4	0	24	0/0	1/1	8/8	4/4	7/7	20/20	44	**14%**	**75%**	17%	83%
Held-out Test Set																	
Total	6	8	6	15	11	46	0/0	1/1	4/4	4/4	7/7	16/16	62	0%		0%	
No Learning	4	7	4	9	5	29	0/0	0/0	2/0	2/0	2/0	6/0	29	53%		100%	
Learning	4	7	4	9	3	27	0/0	0/0	2/2	2/2	2/2	6/6	33	47%	12%	63%	37%

The resulting 12% relative reduction in error rate (Table 7.8, *Total RER* column) for recognition of all label events in the held-out test set was not significant by a McNemar test; however, the 37% absolute error rate reduction for the recognition of abbreviated labels alone (*Abbrev AER* column) was significant (McNemar test, p<=0.03).

Given the definition of learning above, SHACER has demonstrated significant learning on a previously unseen test set. With the ability to (1) make inferential associations across modes and (2) use remembered learning across meetings, SHACER was able to identify the lexical/semantic aspects of Gantt chart labels significantly better than without those abilities. SHACER accomplished this by means of its primary processes: *alignment*, *refinement* and *integration*. These processes are used to unobtrusively detect instances of multimodal redundancy across handwriting and speech. Therefore, SHACER proves the concept that bootstrapped learning from leveraging multimodal redundancy is possible. SHACER can dynamically combine lower-level recognitions into new higher level knowledge (like learned abbreviations) and continually build up its level of understanding.

7.7 SUMMARY

In designing the thresholds and heuristic conditions that supported recognition and learning the aim was to discover plausible and general procedures that were not specific to the development test set. In particular, the techniques of (1) articulatory-feature based alignment and (2) positional bigram modeling to constrain the second pass search over saved speech features are general in this sense.

The tests described in Section 7.5.1 showed that an upper bound for combining phonetic information from letter-to-sound transformed handwriting with phonetic information from speech could be a doubling of phone-level recognition accuracy. Thus, the concept of using handwriting information combined with speech information to improve phone recognition has also been shown to be plausible and generally effective. Using articulatory-feature based distances for aligning the letter-to-sound transformations of the handwriting alternatives with phone ensemble outputs was shown to be useful and effective.

Overall, the cumulative effectiveness of SHACER's general procedures was evaluated by testing on the held-out test data. The held-out test results demonstrated the system's ability to learn new terms, based on these general procedures. Further, tests showed that using these approaches within SHACER made a significant difference — a 37% absolute reduction in recognition error rate, in recognizing the expanded meanings of out-of-vocabulary abbreviations. This showed that SHACER's concept of learning new terms on the basis of detecting and dynamically analyzing instances of multimodal redundancy works. It also showed that SHACER's transfer of learned meanings from newly recognizable symbols or expressions in one mode (e.g. Figure 7.4, *G1*, spoken and handwritten *Joe Browning*) to previously unseen symbols in another mode (e.g. Figure 7.4, *G2*, handwritten abbreviation, *JB*) is feasible and can significantly improve the system's level of understanding.

Chapter 8

Conclusion and Future Work

8.1 CONCLUSION

This dissertation introduced SHACER, a system for recognizing instances of redundant handwriting and speech during common human-human interactions like meetings. There are many large research projects currently studying human interactions during meetings [37, 165, 4, 33, 106, 29, 24], trying to devise ways that computational systems can be a benefit to people in these situations. SHACER's contribution of dynamic vocabulary learning to this arena of computational system development is specific, timely and significant.

8.1.1 Specificity of Contribution

As shown in chapter 2 people typically say what they publicly handwrite during a meeting. SHACER is the first research system to specifically attempt to leverage such instances of multimodal redundancy across handwriting and speech to support dynamic learning and adaptation — for example, by enrolling out-of-vocabulary terms based on handwriting and speech redundancy (Chapter 6) and then using those enrolled terms to better understand subsequent abbreviations (Chapter 7).

8.1.2 Timeliness of Contribution

Speech and handwriting recognizers are closely related, mature in terms of the capabilities and relatively inexpensive, and as such lend themselves particularly well to the support of a system that learns dynamically over time. Our Introduction (Chapter 1) listed possible contexts in which the occurrence of multimodal redundancy could be leveraged for improving recognition and adaptive learning. Tracking deictic glances or gestures (e.g. looking or pointing) and combining such gestures with speech is an exciting area of research for dynamic learning; however, vision-based object recognition is still relatively immature and expensive, whereas handwriting and speech recognizers are common. For example the Tablet PC architecture includes both speech and handwriting recognition by default. Fully handwritten words or phrases are often language-specific iconic representations of spoken pronunciations (e.g., abbreviations and acronyms), and thus support a broad range of straightforward iconic perceptual grounding — i.e., associating orthographic text to the pronunciation of spoken utterances. Therefore, in terms of research into systems that can acquire new language both graphically (via handwriting) and auditorily (via speech) SHACER is timely, and suggests a broad and rich area of immediately available further study — like lecture understanding or collaborative annotation systems in which both spoken and handwritten information can be readily collected.

8.1.3 Significance of Contribution

Finally, the results presented in this thesis are significant. Testing of SHACER (in Chapter 7) has shown that it is possible, by leveraging multimodal redundancy, to transfer meaning from known symbols in one mode to unknown symbols in another mode (Section 6.4). This ability can have a substantial effect in improving background understanding of the natural interactions occurring during meetings. Testing of SHACER yielded a significant 37% reduction in absolute abbreviation recognition error rate.

We have argued that computational perceptual systems in addition to their off-line-trained recognizers will need to observe and learn from previously unseen natural demonstrations. To accomplish this kind of bootstrapped learning and adaptation they will need to assemble the outputs of their recognizers, which have been trained with supervision off-line, into meaningful higher-level symbols in real-time, without supervision. SHACER is a prototype of how to do this by leveraging multimodal redundancy.

8.1.4 Contributions

This dissertation has made research contributions in the following areas.

Multimodality

1. Chapter 2 showed that contrary to prevailing knowledge on modality usage in human-computer, multimodal command interface systems, which show that multimodal redundancy occurs in only 1%-5% of interactions, handwriting and speech redundancy is typical for some computer-mediated, human-human interactions, with redundant speech accompanying handwriting well over 95% of the time.

2. Previously Anderson *et al.* examined redundant speech associated with handwriting during computer-mediated distance lecture delivery. Their study looked at only 54 instances of handwriting. This dissertation examined an order of magnitude more data (688 handwriting instances) across three separate contexts: whiteboard presentations, a spontaneous brainstorming session, and photo annotation/discussion sessions. In these contexts, handwriting occurs on a shared space during human-human interactions, which is also true for Anderson *et al's* study. Our finding was that 96.5% of handwritten words are also spoken redundantly. This finding both complements and extends Anderson *et al's* work. Whereas Anderson *et al's* study was specific to the use of tablet PCs, none of the three scenarios we examined included the use of a tablet PC (Section 2.2.3).

3. In studying a small corpus of multi-party photo annotation discussions (Section 2.2.2) this dissertation offers evidence that handwriting on digital paper during a public discussion is accompanied by redundant speech, just as is handwriting on a whiteboard during a lecture, presentation or brainstorming session.

4. This dissertation offers strong evidence both from our earlier work with Multimodal New Vocabulary Recognition (MNVR — Chapter 4) and from our current work with SHACER that combining information from redundant handwriting and speech leads to better recognition than can be achieved in either mode alone. MNVR test results in Section 4.5.2 show a significant 57.5% relative error rate reduction compared to speech-only pronunciations, and a significant 7.04% relative error rate reduction compared to handwriting-only pronunciations. SHACER test outcomes in Section 7.6 show a significant 37% absolute reduction in error rate compared to either handwriting-only or speech-only recognition of abbreviated Gantt chart labels, because neither handwriting nor speech alone succeeded in producing any correct recognitions for such abbreviated labels.

5. In Section 2.2.3 we have shown that (a) redundantly presented terms are likely to be out-of-vocabulary (OOV) for ideally sized recognition vocabularies, (b) likely to be memorable to meeting participants (by virtue of having been presented redundantly across multiple informative modes — e.g. handwriting and speech), and also (c) likely to be dynamically learnable. We contend that learning these words dynamically could offer a viable alternative to drastically increasing vocabulary size for tasks like lecture and meeting transcription or summarization.

6. This dissertation introduced the working hypothesis that people use multimodal redundancy (e.g. across handwriting and speech) as a conversational strategy to bolster their communicative effectiveness by drawing attention to the meanings of dialogue-critical terms (Section 2.2.2). To begin establishing empirical support for this hypothesis we derived two claims: (1) if multimodal redundancy is a general conversational strategy then it should be typical of human-human interaction settings where multiple modes can be perceived, and (2) if redundantly presented terms are dialogue-critical then they should be measurably more important than other words. Both claims are supported by the evidence. The first claim is supported by the finding of 96.5% redundancy averaged over three interaction settings (Section 2.2.3), which means that multimodal redundancy in some cases is typical. Claim two is supported by the significantly higher *tf-idf* weights associated with redundantly presented terms, which shows that they are measurably more important (Section 2.2.3) than non-redundant terms.

Multimodal System Architecture

1. In Section 3.1.3 a new class of multimodal system was introduced. This ambient-cumulative interface (ACI) system, instead of supporting a direct human-computer interface for sequences of command/display turns, accumulates ambient perceptual observations during structured multi-party interactions like the construction of a Gantt schedule chart during a meeting. SHACER was deployed for within an ACI called Charter. Charter provided a shared work space during Gantt chart schedule meetings, which was the basis for SHACER's development and testing.

2. Section 7.2 showed that our *edge-splitting* technique, which is an extension of the basic temporal chart parsing algorithm for multimodal systems (Section 7.2), resulted in a significant 46.2% relative reduction in multimodal recognition error rate by avoiding the underlying parsing problems related to incorrect turn segmentation. Without this fundamental architectural innovation SHACER could not be successful.

3. Section 4.3.2 introduced our approach to embedding a recursive transition network (RTN) of separate finite-state grammar recognizers into the architecture of CMU's Sphinx 2 continuous speech recognizer (Figure 4.6, and see Appendix B for specific grammars used by SHACER). This two-level RTN structure supported both (1) specifying the location of out-of-vocabulary words within a defined carrier phrase in

MNVR (Figure 4.7, and (2) techniques used in SHACER to provide a basis for the Word/Phrase-Spotting Recognizer (Section B.4). This symbolic grammar-defined approach to two-level recognition is advantageous for new task domains in which there is not yet much data for stochastically modeling the transition between RTN levels.

4. Chapter 5 contributes a description of the low-level algorithmic aspects of integrating handwriting and redundant speech. In particular it describes SHACER's notable augmentations of articulatory-feature based dynamic programming alignment: (a) diphthong expansion (Section 5.3.5), and (b) measuring the *coherence* (Section 5.3.4) of phonetic alignments to rank the possibility of a redundancy match.

5. Section 5.4.1 contributes a viable approach to extracting LVCSR lattice terms based on the temporal boundaries of handwriting and redundant speech alignment. This is a general technique that can be used to improve multimodal recognition regardless of whether the target term is in or out-of-vocabulary, as exemplified in the *Fred Green* example in Section 6.3.1 where the two-word name, *Fred Green*, is not in the speech recognizer's language model. Thus, although both *Fred* and *Green* are individually in the recognizer's language model, recognition of the two-word sequence is deprecated and the recognizer defers to the in-language-model sequence, "Fred's Green." Using the lattice term extraction and the ranking techniques described in Section 5.4.1 this error is corrected.

Learning

1. SHACER successfully demonstrated first that learning new vocabulary dynamically from even a single instance of redundant handwriting and speech can be done.

   Secondly, SHACER proved that after learning has been accomplished, a newly enrolled term can be subsequently recognized in one mode and its meaning transferred to a new, unknown symbol in another mode. This is the basis for SHACER's successful understanding of previously unknown abbreviations.

8.2 FUTURE WORK

8.2.1 Moving Toward Stochastic Decision Processes

In the future as we are able to deploy larger data gathering efforts and formalize our abilities to annotate multimodal perceptual input streams for complex human-human interactions like meetings and presentations, we will move toward statistically modeling the concordances between input modes in order to better weight the confidence we place on various contributions. For example, our current heuristics for deciding how much importance to ascribe to a concordance between a handwriting alternative and an LVCSR transcript term, or between a handwriting alternative and a word/phrase-spotter recognition term are intuitive (see Section 6.2); whereas, with a larger database of annotated examples we could move toward a hybrid symbolic/statistical model such as we employed in our earlier Members, Teams, Committees (MTC) approach for biasing the confidence that the system placed on various input mode contributions [91, 172].

8.2.2 Better Modal Recognizers

SHACER's current base-line phone recognition accuracy rates are under 50%. We know that if handwriting is completely redundant to speech then using oracular phone transcripts, which represent ideal handwriting recognition and letter-to-sound transformation, as part of the ensemble lattice does improve overall recognition (Section 7.5.1), resulting in as much as a doubling of the phone accuracy rate. With better base-line phonetic speech recognition this benefit will increase. Therefore we believe that the task of combining information from handwriting and speech motivates renewed effort in building better phone recognition systems.

Aside from SHACER, other sub-word language processing approaches are now coming into the main stream in areas like spoken document retrieval, call center data mining, and audio/video indexing and searching (e.g. Nexidia [122], Virage [163]). For some years NT-CIR — The Japanese National Institute of Informatics (NII) Test Collection for Information Retrieval (IR) Systems Project — in collaboration with NIST (the National Institute of Standards and Technology) has sponsored a Cross-Lingual Information Retrieval Task (CLIR) track (Nist 2006 Cross-Lingual Information Retrieval Task (Clir)) for searching through spoken documents or telephone speech in any language [156]. In effect this work is developing a universal speech recognition system based on a sub-word unit database covering all the worlds phonemes. This effort is spurred on by the commercial success of language specific versions of sub-word recognition approaches (e.g. Nexidia [122]), which like SHACER's use of phonetic recognition, are better able to handle out-of-vocabulary terms.

SHACER's phone recognition capabilities could be developed to be on par with the phone recognition technology underlying recent successes in spoken document retrieval like those which motivate NIST's effort above to build a langauge universal phonemic recognition system. We believe that leveraging redundant speech and handwriting (because of its clear applications in meeting summarization, lecture transcription, picture and map annotation tracking and understanding, etc.) is another application area ripe for an effort in building fast, efficient and appropriate sub-word recognizers. To move forward in this regard SHACER also needs a better ability to constrain handwriting recognition with information from redundant speech. This would require code-level access to a handwriting recognizer, so that new constraint algorithms like SHACER's dynamic positional bigram modeling (Section 5.4.2) could be developed and applied during a second pass handwriting recognition.

8.2.3 Better Structural Understanding of Syntax and Semantics

Speech and handwriting transcripts could potentially be improved by using speech and handwriting from all participants during meetings. In order to generally align handwritten notes to speech across multiple participants in meetings, presentations and lectures SHACER needs better structural understanding of the syntax and semantics of both the handwriting and speech inputs. This will facilitate a wider and more robust array of matching techniques across not only lexical but also semantic redundancy. The system could be semantically expanded to include information from WordNet [53] or VerbNet [93] synsets, etc.

This effort to better understand the structural relationships between the notes and speech of multiple handwriters and speakers will by necessity be stochastic, which means that to move forward in this regard large scale data collection efforts are probably needed. MIT has demonstrated the advantage of well deployed and long-lived data collection efforts (e.g. Jupiter [186] and lecture transcription systems [60]) for improving spoken language processing. Semantic web-tagging has motivated several deployed games on the internet for gathering data on picture tagging [1, 2]. For acquiring the multimodal data that SHACER needs, one possible arena of data collection could be on-line distributed picture annotation, in which groups of people discuss photos that are mutually visible even to distant participants. The photos can be visible in some locations on computer screens or televisions, in other locations on tablet PCs, and finally also they can printed on digital paper (as in our photo annotation discussions — Section 2.2.2) [16]. Users would be motivated to participate in this scenario because of their own desires to share photos with friends and families. Photos are increasingly recorded and stored digitally, so new ways of sharing photos are needed. Such an online service could potentially be designed to gather both speech and ink data. There would be encumbrances involved (e.g., the need to wear some sort of microphone or sit in front of an array of microphones, and the need to use a tablet PC or digital pen and paper), but these could be outweighed by the benefits and inherent attraction of being able to share photos in new ways. Also SHACER's underlying ability to leverage multimodal redundancy for better understanding could result in useful annotations, which were created "for free" as a side-effect of sharing and conversing about a set of photos. These multimodal annotations, could be stored along with the photos (as is supported in the emerging MPEG-7 standard [116]. As shown in Section 2.2.3 such annotations, which may include a high percentage of redundantly handwritten and spoken words, would make very effective search indexes for organizing and finding photos.

8.2.4 Beyond Handwriting and Speech: MOOVR

Finally, another level of future work involves study and prototyping of systems for other redundant Multimodal Out-Of-Vocabulary Recognition (MOOVR) combinations, for example gaze and speech, point and speech, activity and speech, document context and speech, dynamic learning of sketch constituents, etc. Several research groups are now addressing the possibility of combining these various redundant modal possibilities (e.g. within the CALO project). Work in these other areas of multimodal redundancy could also lead us back into the study of interactions in Virtual/Augmented environments where keyboards and mice are less effective, such as MAVEN [84], which first inspired our thoughts about MOOVR/SHACER.

8.2.5 Better Understanding of Multimodal Redundancy

We don't fully understand why and when people use multimodal redundancy [86]. The literature on early language acquisition suggests the importance of multimodality for focusing attention [61, 12] and thus providing a basis for understanding intentionality and establishing the meaning associations between action and intention that ground language [13, 14, 108, 168, 171, 177, 15]. The Theory of Multimedia Learning [109] explains why students who watch multimedia presentations with embedded multimodal redundancy have enhanced memory recall. However, it is not yet fully clear why a presenter would use handwriting and speech redundantly. Is it a conscious effort to enhance communication, or does it arise unconsciously — for instance, from McNeill's hypothesized speech/gesture growth point [110]?

As we gain understanding of when and why people use redundant handwriting and speech, a related question will be how that tendency could be usefully invoked in certain human-computer interfaces. If multimodal redundancy comes naturally to people in some settings, as shown in Chapter 2, then can machine learning interfaces be designed that encourage people to be redundant? If so, then as we have shown, it is possible to leverage that redundancy for dynamic unsupervised machine learning.

Ultimately we hope and believe our work with SHACER is part of a larger gradual shift that is taking place from computer-centric to human-centric applications. Such applications will begin to shift more and more of the burden of learning and adaptation off of the user and onto the machine.

Bibliography

[1] Ahn, L. v. Games with a purpose. *IEEE Computer Magazine 39*, 6 (2006), 92–94.

[2] Ahn, L. v., Liu, R., and Blum, M. Peekaboom: A game for locating objects in images. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems, ACM CHI 2006* (2006), pp. 55–64.

[3] Allauzen, A., and Gauvain, J.-L. Open vocabulary asr for audiovisual document indexation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '05* (Philadelphia, 2005), vol. 1, pp. 1013–1016.

[4] AMI. *Augmented Multi-party Interaction.* http://www.amiproject.org/ [Viewed: March 10, 2007].

[5] Anderson, R., Hoyer, C., Prince, C., Su, J., Videon, F., and Wolfman, S. Speech, ink and slides: The interaction of content channels. In *Proceedings of the 12th Annual ACM International Conference on Multimedia, ACM Multimedia '04* (2004), pp. 796–803.

[6] Anderson, R. J., Anderson, R., Hoyer, C., and Wolfman, S. A. A study of digital ink in lecture presentation. In *CHI 2004: The 2004 Conference on Human Factors in Computing Systems* (Vienna, Austria, 2004), pp. 567–574.

[7] Arnon, A., Alon, E., and Srinivasan, S. Advances in phonetic word spotting. In *Tenth International Conference on Information and Knowledge Management* (Atlanta, Georgia, 2001), pp. 580–582.

[8] Asadi, A. O. *Automatic Detection and Modeling of New Words in a Large Vocabulary Continuous Speech Recognition System.* Ph.d thesis, Northeastern University, 1991.

[9] Atkeson, C. G., Hale, J. G., Pollick, F., Riley, M., Kotosaka, S., Schaal, S., Shibata, T., Tevatia, G., Ude, A., Vijayakumar, S., and Kawato, M. Using humanoid robots to study human behavior. *IEEE Intelligent Systems 16*, 4 (2000), 46–56.

[10] Baddeley, A. D. *Human Memory: Theory and Practice.* Allyn and Bacon, Boston, 1998.

[11] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval.* Addison-Wesley, 1999.

[12] Bahrick, L., Lickliter, R., and Flom, R. Intersensory redundancy guides infants selective attention, perceptual and cognitive development. *Current Directions in Psychological Science 13* (2004), 99–102.

[13] Baird, J. A., and Baldwin, D. A. Making sense of human behavior: Action parsing and intentional inference. In *Intentions and Intentionality*, B. F. Malle, L. J. Moses, and D. A. Baldwin, Eds. MIT Press, Cambridge, MA., 2001, pp. 193–206.

[14] Baldwin, D., and Baird, J. A. Discerning intentions in dynamic human action. *TRENDS in Cognitive Science 5*, 4 (2001), 171–178.

[15] Baldwin, D. A., Markman, E. M., Bill, B., Desjardins, R. N., Irwin, J. M., and Tidball, G. Infants reliance on a social criterion for establishing word object relations. *Child development 67* (1996), 3125–3153.

[16] Barthelmess, P., Kaiser, E. C., Huang, X., and Demirdjian, D. Distributed pointing for multimodal collaboration over sketched diagrams. In *The Seventh International Conference on Multimodal Interfaces (ICMI '05)* (Trento, Italy, 2005), pp. 10–17.

[17] Barthelmess, P., Kaiser, E. C., Huang, X., McGee, D., and Cohen, P. Collaborative multimodal photo annotation over digital paper. In *Eighth International Conference on Multimodal Interfaces (ICMI06), to appear* (Banff, Canada, 2006).

[18] Bates, R. *Speaker Dynamics as a Source of Pronunciation Variability for Continuous Speech Recognition Models.* PhD thesis, University of Washington, 2003.

[19] Bazzi, I. *Modelling Out-of-Vocabulary Words for Robust Speech Recognition.* Phd. thesis, Massachusetts Institute of Technology, 2002.

[20] Bazzi, I., and Glass, J. R. Modeling out-of-vocabulary words for robust speech recognition. In *Proceedings of the 6th International Conference on Spoken Language Processing* (Beijing, China, 2000), pp. 401–404.

[21] Black, A., Taylor, P., and Caley, R. Technical report hcrc/tr-83, the festival speech synthesis system: System documentation. Tech. rep., Human Communication Research Centre, January 1997.

[22] Black, A. W., and Lenzo, K. A. Flite: a small fast run-time synthesis engine. In *The 4th ITWR ISCA Worskop on Speech Synthesis* (Perthshire, Scotland, 2001).

[23] Bluethmann, W., Ambrose, R. O., Diftler, M., Askew, S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., and Magruder, D. Robonaut: A robot designed to work with humans in space. *Autonomous Robots 14*, 2-3 (2003), 179–197.

[24] Brachman, R., and Lemnios, Z. Darpa's new cognitive systems vision. *Research News 14*, 5 (2002), 1–8.

[25] Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., Lockerd, A., and Mulanda, D. Humanoid robots as cooperative partners for people. *International Journal of Humanoid Robots 2*, 1 (2004).

[26] Breazeal, C., Hoffman, G., and Lockerd, A. Teaching and working with robots as a collaboration. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)* (New York, 2004), vol. 3, pp. 1030–1037.

[27] Brennan, S. Lexical entrainment in spontaneous dialogue. In *In Proceedings of the International Symposium on Spoken Dialogue* (Philadelphia, USA, 1996), pp. 41–44.

[28] Burget, L., Cernock, J., Fapso, M., Karafit, M., Matejka, P., Schwarz, P., Smrz, P., and Szke, I. Indexing and search methods for spoken documents. In *Proceedings of the Ninth International Conference on Text, Speech and Dialogue, TSD '06* (Berlin, DE, 2006), pp. 351–358.

[29] CALO. http://www.ai.sri.com/project/CALO [Viewed: March 10, 2007].

[30] Cangelosi, A., Greco, A., and Harnad, S. From robotic toil to symbolic theft: Grounding transfer from entry-level to higher-level categories. *Connection Science 12* (2000), 143–162.

[31] Cardillo, P. S., Clements, M., and Miller, M. S. Phonetic searching vs. lvscr: How to find what you really want in audio archives. *International Journal of Speech Technology 5* (2002), 9–22.

[32] Cardillo, P. S., Clements, M., and Miller, M. S. Phonetic searching vs. lvcsr: How to find what you really want in audio archives. *International Journal of Speech Technology 5*, 1 (2004), 9–22.

[33] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. The ami meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005.*, S. Renals and S. Bengio, Eds., vol. 3869. Springer-Verlag Lecture Notes in Computer Science Volume, 2005, pp. 28–39.

[34] Carletta, J. C. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics 22*, 2 (1996), 249–254.

[35] Chai, J. Y., Prasov, Z., Blaim, J., and Jin, R. Linguistic theories in efficient multimodal reference resolution: An empirical investigation. In *International Conference on Intelligent User Interfaces* (San Diego, CA, 2005), ACM Press, pp. 43–50.

[36] Charniak, E., and McDermott, D. *Introduction to Artificial Intelligence.* Addison-Wesley Publishing Company, Reading, MA., 1985.

[37] CHIL. *Computers In the Human Interaction Loop.* http://chil.server.de/servlet/is/101/ [Viewed: March 10, 2007].

[38] Chronis, G., and Skubic, M. Sketched-based navigation for mobile robots. In *In Proceedings of the 2003 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2003)* (St. Louis, MO, 2003), pp. 284–289.

[39] Chung, G., Seneff, S., and Wang, C. Automatic acquistion of names using speak and spell mode in spoken dialogue systems. In *Proceedings of HLT-NAACL 2003* (Edmonton, Canada, 2003), pp. 197–200.

[40] Chung, G., Seneff, S., Wang, C., and Hetherington, L. A dynamic vocabulary spoken dialogue interface. In *Interspeech '04* (Jeju Island, Korea, 2004), pp. 321–324.

[41] Chung, G., Wang, C., Seneff, S., FIlisko, E., and Tang, M. Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation. In *Interspeech '04* (Jeju Island, Korea, 2004), pp. 1457–1560.

[42] Clark, H. H. *Using Language.* Cambridge University Press, 1996.

[43] Clark, H. H., and Wilkes-Gibbs, D. Referring as a collaborative process. *Cognition 22* (1986), 1–39.

[44] Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. Quickset: Multimodal interaction for distributed applications. In *International Multimedia Conference* (1997), pp. 31–40.

[45] Cohen, P., McGee, D., Oviatt, S., Wu, L., Clow, J., King, R., Julier, S., and Rosenblum, L. Multimodal interaction for 2d and 3d environments. *IEEE Computer Graphics and Applications 19*, 4 (1999), 10–13.

[46] Cohen, P. R., Cheyer, A. J., Wang, M., and Baeg, S. C. An open agent architecture. In *AAAI Spring Symposium* (1994), pp. 1–8.

[47] Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. Quickset: multimodal interaction for simulation set-up and control. In *Proceedings of the fifth conference on Applied natural language processing* (1997), pp. 20–23.

[48] Cooper, R. M. The control of eye fixation by the meaning of spoken language: A new methodology for the real-time investigation of speech perception, memory, and language processing. *Cognitive Psychology 6* (1974), 84–107.

[49] CSLU. http://cslu.cse.ogi.edu/toolkit/index.html [Viewed: March 10, 2007], CSLU Toolkit, Center for Spoken Language Understanding, OHSU.

[50] Cypher, A., and Halbert, D. C. *Watch What I Do: Uses of Demonstrational Techniques.* MIT Press, 1993.

[51] Demirdjian, D., Ko, T., and Darrell, T. Constraining human body tracking. In *Proceedings of the International Conference on Computer Vision* (Nice, France, 2003), pp. 1071–1078.

[52] Dhande, S. S. *A Computational Model to Connect Gestalt Perception and Natural Language.* Master thesis, Massachesetts Institute of Technology, 2003.

[53] Fellbaum, C., Ed. *WordNet: An Electronic Lexical Database.* MIT Press, Boston, 1998.

[54] Filali, K., and Bilmes, J. A dynamic bayesian framework to model context and memory in edit distance learning: An application to pronunciation classification. In *Proceedings of the Association for Computational Linguistics (ACL)* (University of Michigan, Ann Arbor, 2005), pp. 338–345.

[55] Franklin, D., and Hammond, K. The intelligent classroom: Providing competent assistance. In *In Proceedings of International Comference on Autonomous Agents (Agents-2001)* (2001), pp. 161–168.

[56] Furnas, G., Landauer, T., Gomez, L., and Dumais, S. T. The vocabulary problem in human-system communication. *Communications of the Association for Computing Machinery 30*, 11 (1987), 964–971.

[57] Galescu, L. Technical report of ieice 102(108), sp2002-30: Sub-lexical language models for unlimited vocabulary speech recognition. Invited paper at the symposium on robust spoken language processing systems, organized by the speech committee of the institute of electronics, information and communication engineers and the acoustic society of japan., ATR, May 30-31 2002.

[58] Garofolo, J., Auzanne, G., and Voorhees, E. The trec spoken document retrieval track: A success story. In *RAIO-2000: ContentBased Multimedia Information Access Conference* (Paris, France, 2000), vol. 1, pp. 1–20.

[59] Glaser, R. Education and thinking: The role of knowledge (technical report pds-6). Tech. rep., University of Pittsburgh, Learning Research and Development Center, June 1983.

[60] Glass, J., Hazen, T. J., Hetherington, L., and Wang, C. Analysis and processing of lecture audio data: Preliminary investigations. In *HLT-NAACL Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval* (2004), pp. 9–12.

[61] Gogate, L. J., Walker-Andrews, A. S., and Bahrick, L. E. The intersensory origins of word comprehension: an ecological-dynamic systems view. *Development Science 4*, 1 (2001), 1–37.

[62] Gorniak, P., and Roy, D. K. Augmenting user interfaces with adaptive speech commands. In *In Proceedings of the International Conference for Multimodal Interfaces* (Vancouver, B.C., Canada, 2003), pp. 176–179.

[63] Grice, H. P. Logic and conversation. In *Speech Acts*, P. Cole and J. Morgan, Eds. Academic Press, New York, 1975, pp. 41–58.

[64] Gu, H., Li, J., Walter, B., and Chang, E. Spoken query for web search and navigation. In *Poster Proceedings, Tenth International World-Wide Web Conference* (Available: http://www10.org/cdrom/posters/1010.pdf [Viewed: March 10, 2007], 2001).

[65] Gupta, A. K., and Anastasakos, T. Dynamic time windows for multimodal input fusion. In *Proceedings of INTERSPEECH-2004* (Jeju Island, Korea, 2004), pp. 1009–1012.

[66] Gupta, A. K., and Anastasakos, T. Integration patterns during multimodal interaction. In *Proceedings of INTERSPEECH-2004* (Jeju Island, Korea, 2004), pp. 2293–2296.

[67] Hansen, J. H., Huang, R., Mangalath, P., Zhou, B., Seadle, M., and John. R. Deller, J. Speechfind: Spoken document retrieval for a national gallery of the spoken word. In *Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004* (Espoo, Finland, 2004), pp. 712–730.

[68] Harnad, S. The symbol grounding problem. *Physica D 42* (1990), 335–346.

[69] Hetherington, I. *A Characterization of the Problem of New, Out-of-Vocabulary Words in Continuous-Speech Recognition and Understanding.* Phd thesis, Massachusetts Institute of Technology, 1995.

[70] Hoffman, G., and Breazeal, C. Robots that work in collaboration with people. In *Proceedings of AAAI Fall Symposium on the Intersection of Cognitive Science and Robotics* (Washington, D.C., 2004), pp. 25–30.

[71] Hosom, J.-P. *Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information.* Phd. thesis, Oregon Graduate Institute, 2000.

[72] Hsieh, Y.-c., Huang, Y.-t., Wang, C.-c., and Lee, L.-s. Improved spoken document retrieval with dynamic key term lexicon and probabilistic latent semantic analysis (plsa). In *Proceedings of ICASSP '06* (2006), vol. 1, pp. 961–964.

[73] Huang, X., and Oviatt, S. Combining user modeling and machine learning to predict users multimodal integration patterns. In *Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006.*, S. Renals, S. Bengio, and J. G. Fiscus, Eds., Lecture Notes in Computer Science. Bethseda, MD, 2006, pp. 50–62.

[74] Itoh, Y. A matching algorithm between arbitrary sections of two speech data sets for speech retrieval. In *Proceedings of ICASSP '01* (Salt Lake City, Utah, 2001), pp. 593–596.

[75] Itoh, Y. Shift continuous dp: A fast matching algorithm between arbitrary parts of two time-sequence data sets. *Systems and Computers in Japan 36*, 10 (2005), 43–53.

[76] Itoh, Y., Tanaka, K., and Lee, S.-W. An algorithm for similar utterance section extraction for managing spoken documents. *Multimedia Systems 10*, 5 (2005), 432–443.

[77] Johnston, M. Unification-based multimodal parsing. In *Proceedings of COLING-ACL 98: The 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (Montreal, Canada, 1998), pp. 624–630.

[78] Johnston, M., and Bangalore, S. Matchkiosk: A multimodal interactive city guide. In *Association of Computational Linguistics (ACL2004)* (Barcelona, Spain, 2004), pp. 128–131.

[79] Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., and Smith, I. Unification-based multimodal integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (1997), Association for Computational Linguistics Press, pp. 281–288.

[80] Jones, G. J. F., Foote, J. T., Jones, K. S., and Young, S. J. Retrieving spoken documents by combining multiple index sources. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Zurich, Switzerland, 1996), pp. 30–38.

[81] Kaiser, E. Robust, finite-state parsing for spoken language. In *Proceedings of the Student Session of the Association for Computational Linguistics, ACL '99* (College Park, Maryland, 1999), pp. 573–578.

[82] Kaiser, E., Demirdjian, D., Gruenstein, A., Li, X., Niekrasz, J., Wesson, M., and Kumar, S. Demo: A multimodal learning interface for sketch, speak and point creation of a schedule chart. In *International Conference on Multimodal Interfaces (ICMI '04)* (State College, PA., 2004), pp. pgs. 329–330.

[83] Kaiser, E., Johnston, M., and Heeman, P. A. Profer: Predictive, robust finite-state parsing for spoken language. In *Proceedings of ICASSP* (1999), vol. 2, pp. 629–632.

[84] Kaiser, E., Olwal, A., McGee, D., Benko, H., Corradini, A., Li, X., Cohen, P., and Feiner, S. Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In *International Conference on Mutimodal Interfaces (ICMI '03)* (2003), pp. 12–19.

[85] Kaiser, E. C. Multimodal new vocabulary recognition through speech and handwriting in a whiteboard scheduling application. In *Proceedings of the International Conference on Intelligent User Interfaces* (San Diego, CA., 2005), pp. 51–58.

[86] Kaiser, E. C. Can modeling redundancy in multimodal, multi-party tasks support dynamic learning? CHI 2005 Workshop: CHI Virtuality 2005.

[87] Kaiser, E. C. Shacer: a speech and handwriting recognizer. In *Workshop Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005), Workshop on Multimodal, Multiparty Meeting Processing* (Trento, Italy, Available: http://www.idiap.ch/ICMI05/programpapers/document_view [Viewed: March 10, 2007], 2005).

[88] Kaiser, E. C., and Barthelmess, P. Edge-splitting in a cumulative multimodal system, for a no-wait temporal threshold on information fusion, combined with an under-specified display. In *Ninth International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP)* (Pittsburgh, PA, Available: http://www.isca-speech.org/archive/interspeech_2006/i06_2016.html [Viewed: March 10, 2007], 2006).

[89] Kaiser, E. C., Barthelmess, P., and Arthur, A. Multimodal play back of collaborative multiparty corpora. In *Workshop Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005), Workshop on Multimodal, Multiparty Meeting Processing* (Trento, Italy, Available: http://www.idiap.ch/ICMI05/programpapers/document_view [Viewed: March 10, 2007], 2005).

[90] Kaiser, E. C., Barthelmess, P., Huang, X., and Demirdjian, D. A demonstration of distributed pointing and referencing for multimodal collaboration over sketched diagrams. In *Workshop Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005), Workshop on Multimodal, Multiparty Meeting Processing* (Trento, Italy, Available: http://caloproject.sri.com/publications/distpointingdemo.html [Viewed: March 10, 2007], 2005).

[91] Kaiser, E. C., and Cohen, P. R. Implementation testing of a hybrid symbolic/statistical multimodal architecture. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002)* (Denver, 2002), pp. 173–176.

[92] Kara, L. B., and Stahovich, T. F. An image-based trainable symbol recognizer for sketch-based interfaces. *Computers and Graphics 29*, 4 (2005), 501–517.

[93] Kipper, K., Dang, H. T., and Palmer, M. Class-based construction of a verb lexicon. In *AAAI-2000 Seventeenth National Conference on Artificial Intelligence* (Austin, Texas, 2000), pp. 691–696.

[94] Ko, T., Demirdjian, D., and Darrell, T. Untethered gesture acquistion and recognition for a multimodal conversational system. In *Fifth International Conference on Multimodal Interfaces, ICMI '03* (Vancouver, B.C., Canada, 2003), pp. 147–150.

[95] Kondrak, G. Alignment of phonetic sequences. Technical report CSRG-402, Technical report CSRG-402, Department of Computer Science, University of Toronto, December 1999.

[96] Kondrak, G. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL)* (Seattle, WA., 2000), pp. 288–295.

[97] Kondrak, G., and Sherif, T. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *COLING-ACL* (Sydney, Australia, 2006), pp. 43–50.

[98] Kumar, S., Cohen, P. R., and Levesque, H. J. The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams. In *In Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000)* (Boston, MA., USA., 2000), pp. 159–166.

[99] Kurihara, K., Goto, M., Ogata, J., and Igarashi, T. Speech pen: Predictive handwriting based on ambient multimodal recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montral, Qubec, Canada, 2006), pp. 851 – 860.

[100] Landay, J. A., and Myers, B. A. Sketching interfaces: Toward more human interface design. *IEEE Computer 34*, 3 (2001), 56–64.

[101] Larkey, L., AbdulJaleel, N., and Connell, M. What's in a name?: Proper names in arabic cross language information retrieval. Tech. rep., Technical Report, IR-278, Center for Intelligent Information Retrieval (CIIR), 2003.

[102] Leath, T. *Audient: An Acoustic Search Engine.* First Year Report, (PhD. program). Available: http://www.infm.ulst.ac.uk/∼ted/exassets/confirmation/Confirmation.pdf, [Viewed March 11, 2007], University of Ulster, 2005.

[103] Levenshtein, V. I. Binary codes capable of correcting spurious insertions and deletions of ones (original in russian). *Russian Problemy Peredachi Informatsii 1* (1965), 12–25.

[104] Lopresti, D., and Wilfong, G. Cross-domain approximate string matching. In *Proceedings of Sixth International Symposium on String Processing and Information Retrieval* (1999), pp. 120–127.

[105] Lopresti, D., and Wilfong, G. Cross-domain searching using handwritten queries. Proceedings of Seventh International Workshop on Frontiers in Handwriting Recognition.

[106] M4. *Multimodal Meeting Manager*. http://www.m4project.org/ [Viewed: March 11, 2007].

[107] MacEachren, A. M., Cai, G., McNeese, M., Sharma, R., and Fuhrmann, S. Geocollaborative crisis management: designing technologies to meet real-world needs. In *Proceedings of the 2006 national conference on Digital government research* (San Diego, California, 2006), pp. 71 – 72.

[108] Malle, B. F., Moses, L. J., and Baldwin, D. A. Introduction: The significance of intentionality. In *Intentions and Intentionality: Foundations of Social Cognition*, B. F. Malle, L. J. Moses, and D. A. Baldwin, Eds. MIT Press, Cambridge, Mass., 2001, pp. 1–27.

[109] Mayer, R. E., and Moreno, R. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist 38*, 1 (2003), 43–52.

[110] McNeill, D. Growth points, catchments, and contexts. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society 7*, 1 (2000).

[111] Meliani, R. E., and O'Shaughnessy, D. New efficient fillers for unlimited word recognition and keyword spotting. In *Proceedings of ICSLP '96* (Philadelphia, Pennsylvania, USA, 1996), vol. 2, pp. 590–593.

[112] Meurville, E., and Leroux, D. D1.2 collection and annotation of meeting room data (part of m4 project: Multimodal meeting manager). Public deliverable, 2004.

[113] Miller, D., Schwartz, R., and Weischedel, R. Named entity extraction from broadcast news. Proceedings of DARPA Broadcast News Workshop.

[114] Mooney, R. J. Learning language from perceptual context: A challenge problem for ai. Proceedings of the 2006 AAAI Fellows Symposium.

[115] Moreau, N., Jin, S., and Sikora, T. Comparison of different phone-based spoken document retrieval methods with text and spoken queries. In *Interspeech'2005-Eurospeech* (Lisbon, Portugal, 2005), pp. 641–644.

[116] Moreau, N., Kim, H., and Sikora, T. Phonetic confusion based document expansion for spoken document retrieval. In *Proceedings of the 23rd annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2004), pp. 81–87.

[117] Moreno, R., and Mayer, R. E. Verbal redundancy in multimedia learning: When reading helps listening. *Journal of Educational Psychology 94*, 1 (2002), 156–163.

[118] Morgan, N., Baron, D., Bhagat, S., Carvey, H., Dhillon, R., Edwards, J., Gelbart, D., Janin, A., Krupski, A., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C. Meetings about meetings: Research at icsi on speech in multiparty conversations. In *Proceedings of ICASSP-'03* (Hong Kong, 2003), vol. 4, pp. 740–743.

[119] Muslea, I., Minton, S., and Knoblock, C. Active + semi-supervised learning = robust multi-view learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)* (2002), pp. 435–442.

[120] Natalie, R., Ronnie, T., and Chen, F. Examining the redundancy of multimodal input. In *Proceedings of OZCHI 2006* (Sydney, Australia, 2006), pp. 389–392.

[121] Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., and Vergyri, D. Large-vocabulary audio-visual speech recognition: A summary of the johns hopkins summer 2000 workshop. In *Proc. IEEE Workshop on Multimedia Signal Processing* (Cannes, 2001), pp. 619–624.

[122] Nexidia. *White Papers.* http://www.nexidia.com/technology/whitepapers.html [Viewed: March 11, 2007].

[123] Ng, C., and Zobel, J. Speech retrieval using phonemes with error correction. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (Melbourne, Australia, 1998), pp. 365 – 366.

[124] Ng, K. Towards robust methods for spoken document retrieval. In *Proceedings of ICSLP '98* (Sydney, Australia, 1998), vol. 3, pp. 939–942.

[125] Ng, K., and Zue, V. Subword-based approaches for spoken document retrieval. *Speech Communication 32*, 3 (2000), 157–186.

[126] Nicolescu, M., and Mataric, M. J. Natural methods for robot task learning: Instructive demonstration, generalization and practice. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems* (Melbourne, Australia, 2003), pp. 241–248.

[127] Nuance. *Dragon AudioMining.* http://www.nuance.com/audiomining/sdk/ [Viewed: March 11, 2007].

[128] Ohtsuki, K., Hiroshima, N., Oku, M., and Imamura, A. Unsupervised vocabulary expansion for automatic transcription of broadcast news. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05)* (Philadelphia, 2005), pp. 1021–1024.

[129] Oviatt, S. Multimodal interfaces for dynamic interactive maps. In *in Proceedings of Conference on Human Factors in Computing Systems* (1996), CHI '96:, New York, ACM Press, pp. 95–102.

[130] Oviatt, S. Integration and synchronization of input modes during multimodal human com-puter interaction. In *Proceedings of CHI* (1997), pp. 415–422.

[131] Oviatt, S. Ten myths of multimodal interaction. *Communications of the ACM 42*, 11 (1999), 74–81.

[132] Oviatt, S., Barthelmess, P., Lunsford, R., and Kaiser, E. Human-centered design of high-performance multimodal interfaces. *IEEE Computer Magazine, Special Issue on Human-centered Computing* (to appear).

[133] Oviatt, S., and Cohen, P. Multimodal interfaces that process what comes naturally. *Communications of the ACM 43*, 3 (2000), 45–53.

[134] Oviatt, S., Coulston, R., Tomko, S., Xiao, B., Lunsford, R., Wesson, M., and Carmichael, L. Toward a theory of organized multimodal integration patterns during human-computer interaction. In *Proceedings of the 5th IEEE International Conference on Multimodal Interfaces (ICMI'03)* (Vancouver, BC, Canada, 2003), pp. 44–47.

[135] Oviatt, S. L. Mutual disambiguation of recognition errors in a multimodal architecture. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (1999), A. Press, Ed., pp. 576–583.

[136] Oviatt, S. L., DeAngeli, A., and Kuhn, K. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '97* (New York:, 1997), ACM Press, pp. 415–422.

[137] Padmanabhan, M., Ramaswamy, G., Ramabhadran, B., Gopalakrishnan, P. S., and Dunn, C. Voicemail corpus part i (ldc98s77) and part ii (ldc2002s35), Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC98S77, Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002S35 [Viewed: March 11, 2007], 1998.

[138] Palmer, D., and Ostendorf, M. Improving out-of-vocabulary name resolution. *Computer Speech and Language 19*, 1 (2005), 107–128.

[139] Pan, S., Shen, S., Zhou, M. X., and Houck, K. Two-way adaptation for robust input interpretation for practical multimodal interaction. In *Proceedings of ACM Conference on Intelligent User Interfaces (IUI)* (2005), pp. 25–32.

[140] Pantic, M., Pentland, A., Nijholt, A., and Huang, T. Human computing and machine understanding of human behavior: A survey. In *Eighth International Conference on Multimodal Interfaces (ICMI '06)* (Banff, Alberta, Canada, 2006), pp. 239–248.

[141] Park, A., and Glass, J. R. Towards unsupervised pattern discovery in speech. In *Proc. ASRU* (San Juan, Puerto Rico, 2005), pp. 53–58.

[142] Park, A., and Glass, J. R. Unsupervised word acquisition from speech using pattern discovery. In *Proceedings of ICASSP '06* (Toulouse, France, 2006), vol. 1, pp. 409–412.

[143] Pook, P. K., and Ballard, D. H. Deictic teleassistance. In *Proc. IEEE/RSJ/GI Int'l Conf. on Intelligent Robots and Systems* (Muenchen, Germany, 1994), pp. 245–252.

[144] Porzel, R., and Strube, M. Towards context-adaptive natural language processing systems. In *Computational Linguistics for the New Millenium: Divergence or Synergy*, M. Klenner and H. Visser, Eds. Peter Lang Academic Publishers, Lang, Frankfurt am Main, Available: http://citeseer.ist.psu.edu/753353.html [Viewed: March 11, 2007], 2002.

[145] Ravishankar, M. Efficient algorithms for speech recognition. Ph.D Thesis Tech Report. CMU-CS-96-143, Carnegie Mellon University, May 1996.

[146] Rosenfeld, R. Optimizing lexical and n-gram coverage via judicious use of linguistic data. In *Eurospeech '95* (1995), pp. 1763–1766.

[147] Roy, D. Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language 16* (2002), 353–385.

[148] Roy, D. Grounded spoken language acquisition: Experiments in word learning. *IEEE Transactions on Multimedia. 5*, 2 (2003), 197–209.

[149] Roy, D., and Mukherjee, N. Towards situated speech understanding: Visual context priming of language models. *Computer Speech and Language 19*, 2 (2005), 227–248.

[150] Roy, D., and Pentland, A. Learning words from sights and sounds: A computational model. *Cognitive Science 26*, 1 (2002), 113–146.

[151] Salton, G., and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management 24*, 5 (1988), 513–523.

[152] Sankoff, D., and Kruskal, J. B. *Time warps, string edits, and macromolecules : the theory and practice of sequence comparison.* CSLI Publiciations, 1999.

[153] Saraclar, M., and Sproat, R. Lattice-based search for spoken utterance retrieval. In *HLT/NAACL* (Boston, 2004), pp. 129–136.

[154] Saund, E., and Mahoney, J. Perceptual support of diagram creation and editing. In *Diagrams 2004 - International Conference on the Theory and Applications of Diagrams* (Cambridge, England, 2004), pp. 424–427.

[155] Schimke, S., Vogel, T., Vielhauer, C., and Dittmann, J. Integration and fusion aspects of speech and handwriting media. In *in Proceedings of the Ninth International Speech and Computer Conference (SPECOM'2004)* (2004), pp. 42–46.

[156] Schone, P., McNamee, P., Morris, G., Ciany, G., and Lewis, S. Searching conversational telephone speech in any of the world's languages. International Conference on Intelligence Analysis.

[157] Seekafile. http://www.seekafile.org/ [Viewed: March 11, 2007].

[158] Seide, F., Yu, P., Ma, C., and Chang, E. Vocabulary-independent search in spontaneous speech. In *Proceedings of ICASSP '04* (Montreal, Canada, 2004), pp. 253–256.

[159] Sethy, A., Narayanan, S., and Parthasarthy, S. A syllable based approach for improved recognition of spoken names. In *Proceedings of the ISCA Pronunciation Modeling Workshop* (Estes Park, CO., 2002), pp. 30–35.

[160] Steil, J. J., Rthling, F., Haschke, R., and Ritter, H. Learning issues in a multi-modal robot-instruction scenario. In *Workshop on Programming by Demonstration, Proceedings of IROS* (Available: http://www.techfak.uni-bielefeld.de/ags/ni/publications/media/SteilRoethlingHaschkeRitter2003-LII.pdf [Viewed: March 11, 2007], 2003).

[161] Szoke, I., Schwarz, P., Matejka, P., Burget, L., Karafiat, M., Fapso, M., and Cernocky, J. Comparison of keyword spotting approaches for informal continuous speech. In *Interspeech'2005 - Eurospeech* (Lisbon, 2005), pp. 633–636.

[162] Tenenbaum, J. B., and Xu, F. Word learning as bayesian inference. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society* (Available: http://citeseer.ist.psu.edu/tenenbaum00word.html [Viewed: March 11, 2007], 2000).

[163] Virage. *Case Studies.* http://www.virage.com/content/downloads/ [Viewed: March 11, 2007], 2006.

[164] Voyles, R. M., and Khosla, P. K. Gesture-based programming: A preliminary demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)* (1999), pp. 708–713.

[165] Waibel, A., Steusloff, H., and Stiefelhagen, R. Chil: Computers in the human interaction loop. In *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)* (Lisbon, Portugal, 2004).

[166] Ward, W. Extracting information from spontaneous speech. In *Proceedings of ICSLP-94* (1994), pp. 83–86.

[167] WaveSurfer. http://www.speech.kth.se/wavesurfer/ [Viewed: March 11, 2007], Department of Speech, Music and Hearing, Royal Institute of Technology (KTH).

[168] Welleman, H. M., and Phillips, A. T. Developing intentional understandings. In *Intentions and Intentionality: Foundations of Social Cognition*, B. F. Malle, L. J. Moses, and D. A. Baldwin, Eds. MIT Press, Cambridge, Mass, 2001, pp. 125–148.

[169] Wermter, S., Elshaw, M., Weber, C., Panchev, C., and Erwin, H. Towards integrating learning by demonstration and learning by instruction in a multimodal robotics. In *IROS-2003 Workshop on Robot Learning by Demonstration* (Las Vegas, Nevada, USA., Available: http://citeseer.ist.psu.edu/677395.html [Viewed: March 11, 2007], 2003).

[170] Wickens, C. D., Sandry, D. L., and Vidulich, M. Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors 25* (1983), 227–248.

[171] Woodward, A. L., Sommerville, J. A., and Guajardo, J. J. How infants make sense of intentional action. In *Intentions and Intentionality*, B. F. Malle, L. J. Moses, and D. A. Baldwin, Eds. MIT Press, Cambridge, MA, 2001, pp. 149–170.

[172] Wu, L., Oviatt, S. L., and Cohen, P. R. From members to teams to committee: A robust approach to gestural and multimodal recognition. *IEEE Transactions on Neural Networks 13*, 4 (Special issue on "Intelligent Multimedia processing") (2002).

[173] Yazgan, A., and Saraclar, M. Hybrid language models for out of vocabulary word detection in large vocabulary conversational speech recognition. In *ICASSP '04* (2004), vol. 1, pp. 745–748.

[174] Young, S. J., Brown, M. G., Foote, J. T., Jones, G. L. F., and Jones, K. S. Acoustic indexing for multimedia retrieval and browsing. In *Proceedings of ICASSP '97* (1997), pp. 199–201.

[175] Yu, C., and Ballard, D. H. A computational model of embodied language learning. Tech. Rep. Technical Report 791, Computer Science Deptartment, University of Rochester, January 2003.

[176] Yu, C., and Ballard, D. H. A multimodal learning interface for grounding spoken language in sensory perceptions. In *International Conference on Multimodal Interfaces (ICMI '03)* (Vancouver, B.C., Canada, 2003), ACM Press, pp. 164–171.

[177] Yu, C., Ballard, D. H., and Aslin, R. N. The role of embodied intention in early lexical acquisition. *Cognitive Science(CogSci 2003) 29*, 6 (2005), 961–1005.

[178] Yu, H., Hripcsak, G., and Friedman, C. Mapping abbreviations to full forms in biomedical articles. *Journal of the American Medical Informatics Association 9* (2002), 262–272.

[179] Yu, H., Tomokiyo, T., Wang, Z., and Waibel, A. New developments in automatic meeting transcription. In *Proceedings of ICSLP 2000* (Beijing, China, Available: http://citeseer.ist.psu.edu/yu00new.html [Viewed: March 11, 2007], 2000).

[180] Yu, P., Chen, K., Lu, L., and Seide, F. Searching the audio notebook: Keyword search in recorded conversation. In *HLT/EMNLP* (2005), pp. 947–954.

[181] Yu, P., Chen, K., Ma, C., and Seide, F. Vocabulary-independent indexing of spontaneous speech. *IEEE Transactions on Speech and Audio Processing 13*, 5 (2005), 635– 643.

[182] ZDNet. *At the Whiteboard*. http://news.zdnet.com/2036-2_22-6035716.html [Viewed: March 11, 2007].

[183] Zhou, B., and Hansen, J. Speechfind: an experimental on-line spoken document retrieval system for historical audio archives. In *ICSLP-2002* (Denver, CO. USA, 2002), vol. 3, pp. 1969–1972.

[184] Zhou, M. X., Wen, Z., and Aggarwal, V. A graph-matching approach to dynamic media allocation in intelligent multimedia interfaces. In *International Conference on Intelligent User Interfaces* (San Diego, CA, 2005), ACM Press, pp. 114–121.

[185] Zhou, Z., Yu, P., Chelba, C., and Seide, F. Towards spoken document retrieval for the internet: Lattice indexing for large scale web search architectures. In *Human Language Technology Conference / North American chapter of the Association for Computational Linguistics Annual Meeting* (New York City, 2006), pp. 415–422.

[186] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T. J., and Hetherington, L. Jupiter: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing 8*, 1 (2000).

Appendix A

Articulatory Features and Phonetic Distances

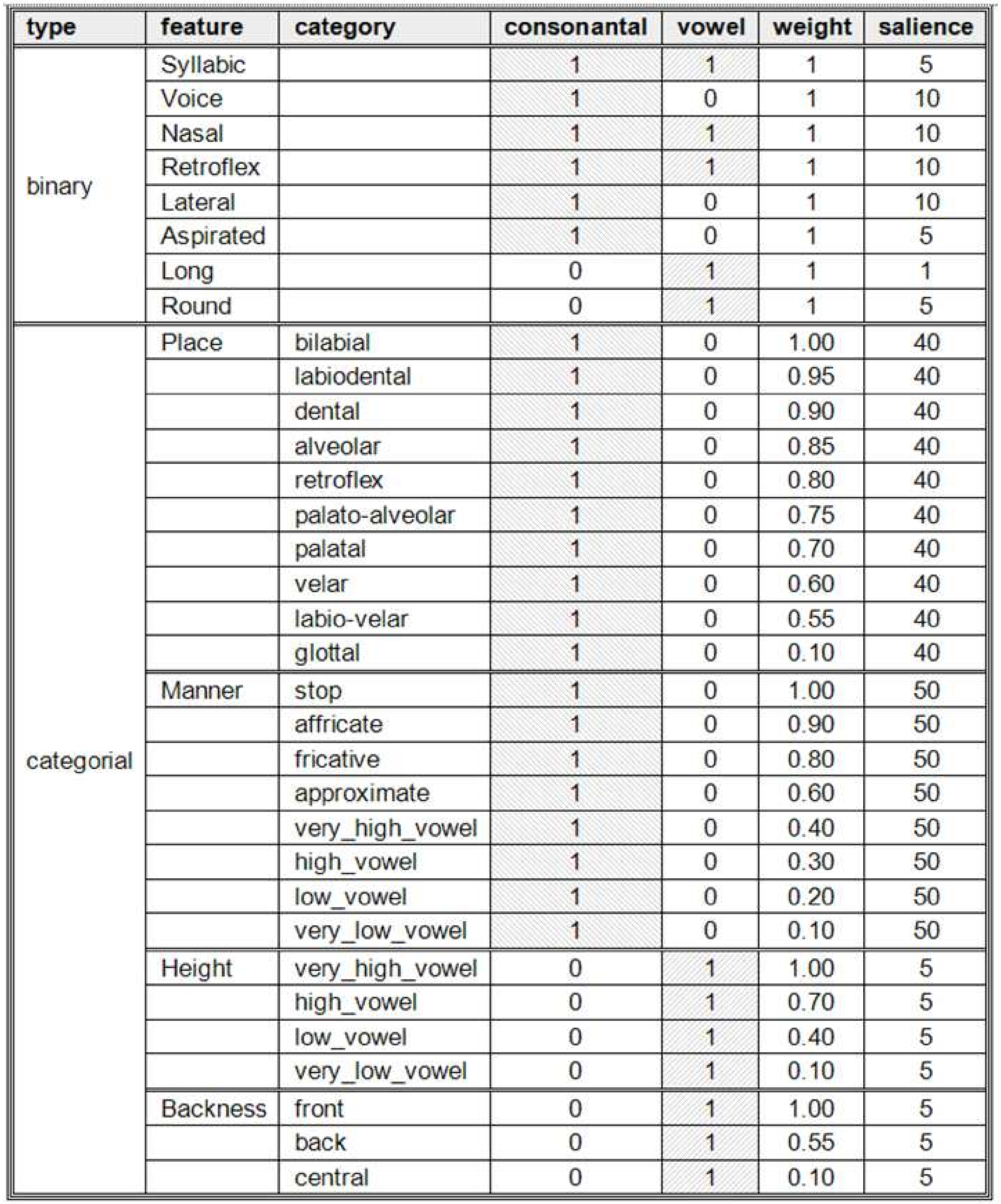
A.1 ARTICULATORY FEATURE TABLE AND PSEUDO-CODE

To align speech and handwriting we have adopted a phonetic alignment approach put forward by Kondrak [96] for articulatory-feature based alignment. Kondrak's algorithm characterizes each phoneme by its set of articulatory features. Some features are binary and some are categorical as shown in Table A.1.

A.1.1 Articulatory Feature Table

All features in Table A.1 have an associated salience weight. Vowels and consonants have different sets of active features. The weight of each categorial feature of each major type (e.g. manner, height, backness, place) is based on empirical linguistic measurements. SHACER does not use all of Kondrak's place features because we are presently only concerned with English phone sequences (e.g. we do not use the features *uvular* and *pharyngeal*). We have augmented Kondrak's height feature to utilize four rather than his original three sub-categories (e.g. Kondrak's original three categories are *high, mid, low*), and in parallel with that we have added a fourth vowel type to the manner feature (in this we follow Hosom [71]).

Table A.1: The table of articulatory features used by SHACER. Some are binary, while others are categorial. Vowels and Consonants have different applicable features, and each feature has a salience weight taken primarily from Kondrak [96].

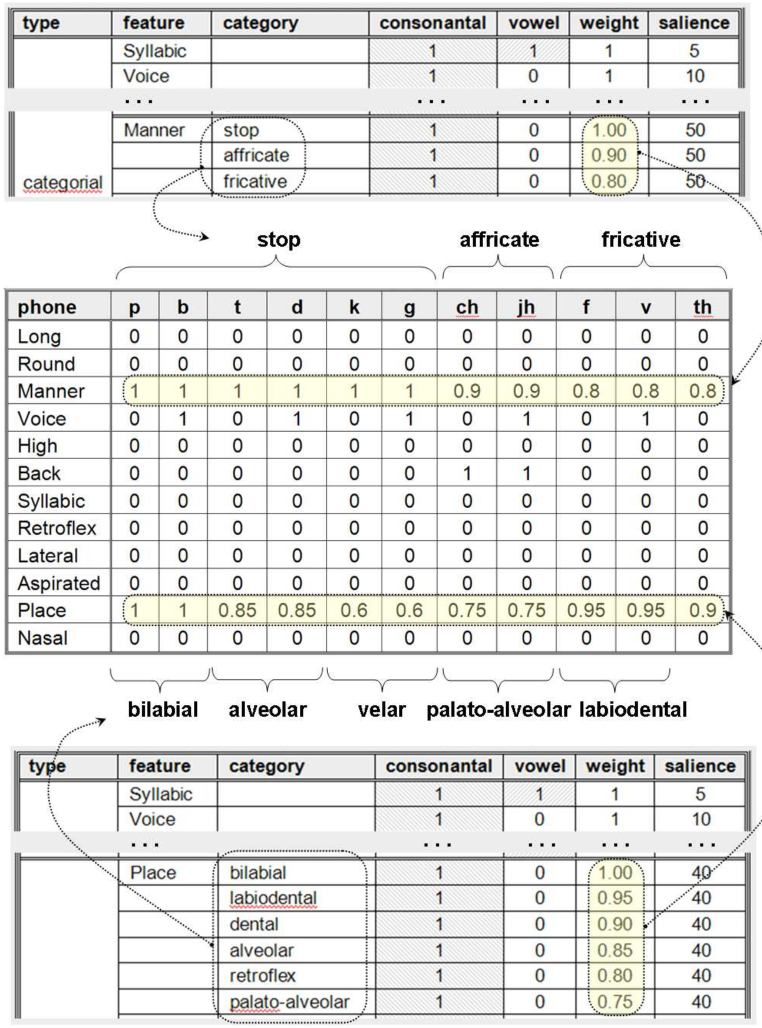


type	feature	category	consonantal	vowel	weight	salience
binary	Syllabic		1	1	1	5
	Voice		1	0	1	10
	Nasal		1	1	1	10
	Retroflex		1	1	1	10
	Lateral		1	0	1	10
	Aspirated		1	0	1	5
	Long		0	1	1	1
	Round		0	1	1	5
categorial	Place	bilabial	1	0	1.00	40
		labiodental	1	0	0.95	40
		dental	1	0	0.90	40
		alveolar	1	0	0.85	40
		retroflex	1	0	0.80	40
		palato-alveolar	1	0	0.75	40
		palatal	1	0	0.70	40
		velar	1	0	0.60	40
		labio-velar	1	0	0.55	40
		glottal	1	0	0.10	40
	Manner	stop	1	0	1.00	50
		affricate	1	0	0.90	50
		fricative	1	0	0.80	50
		approximate	1	0	0.60	50
		very_high_vowel	1	0	0.40	50
		high_vowel	1	0	0.30	50
		low_vowel	1	0	0.20	50
		very_low_vowel	1	0	0.10	50
	Height	very_high_vowel	0	1	1.00	5
		high_vowel	0	1	0.70	5
		low_vowel	0	1	0.40	5
		very_low_vowel	0	1	0.10	5
	Backness	front	0	1	1.00	5
		back	0	1	0.55	5
		central	0	1	0.10	5

A.2 PER PHONE ARTICULATORY FEATURE VALUE TABLE

A listing of the actual articulatory feature values used for each phone in the current SHACER phone set is given in Tables A.2, A.3, A.4, and A.5. The values are assigned as illustrated in Table A.2.

Table A.2: Articulatory feature values for individual phones (p-th). The actual values are assigned based on the categories into which each phone in the phone set is classed; for example, the *stop* phone category all get a *manner* score of *1.0*, while the *affricate* category all get a *manner* score of *0.9*.

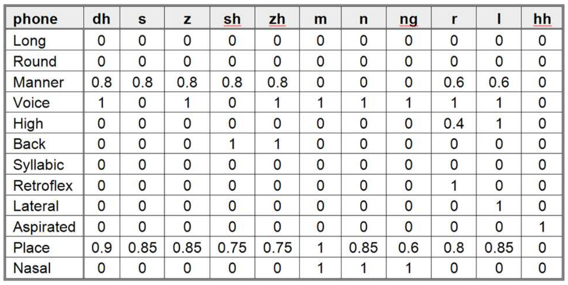


type	feature	category	consonantal	vowel	weight	salience
	Syllabic		1	1	1	5
	Voice		1	0	1	10
	...		...	...	...	...
	Manner	stop	1	0	1.00	50
		affricate	1	0	0.90	50
categorial		fricative	1	0	0.80	50

phone	p	b	t	d	k	g	ch	jh	f	v	th
Long	0	0	0	0	0	0	0	0	0	0	0
Round	0	0	0	0	0	0	0	0	0	0	0
Manner	1	1	1	1	1	1	0.9	0.9	0.8	0.8	0.8
Voice	0	1	0	1	0	1	0	1	0	1	0
High	0	0	0	0	0	0	0	0	0	0	0
Back	0	0	0	0	0	0	1	1	0	0	0
Syllabic	0	0	0	0	0	0	0	0	0	0	0
Retroflex	0	0	0	0	0	0	0	0	0	0	0
Lateral	0	0	0	0	0	0	0	0	0	0	0
Aspirated	0	0	0	0	0	0	0	0	0	0	0
Place	1	1	0.85	0.85	0.6	0.6	0.75	0.75	0.95	0.95	0.9
Nasal	0	0	0	0	0	0	0	0	0	0	0

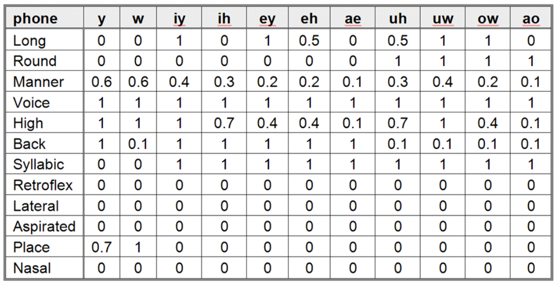
type	feature	category	consonantal	vowel	weight	salience
	Syllabic		1	1	1	5
	Voice		1	0	1	10
	...		...	...	...	...
	Place	bilabial	1	0	1.00	40
		labiodental	1	0	0.95	40
		dental	1	0	0.90	40
		alveolar	1	0	0.85	40
		retroflex	1	0	0.80	40
		palato-alveolar	1	0	0.75	40

Table A.3: Articulatory feature values for individual phones (dh-hh).



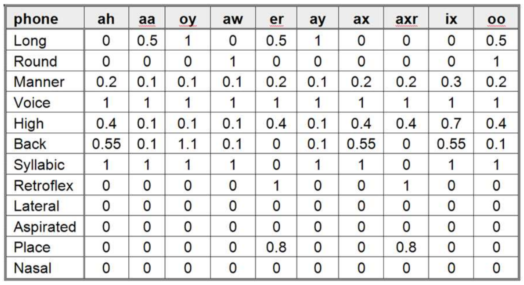
phone	dh	s	z	sh	zh	m	n	ng	r	l	hh
Long	0	0	0	0	0	0	0	0	0	0	0
Round	0	0	0	0	0	0	0	0	0	0	0
Manner	0.8	0.8	0.8	0.8	0.8	0	0	0	0.6	0.6	0
Voice	1	0	1	0	1	1	1	1	1	1	0
High	0	0	0	0	0	0	0	0	0.4	1	0
Back	0	0	0	1	1	0	0	0	0	0	0
Syllabic	0	0	0	0	0	0	0	0	0	0	0
Retroflex	0	0	0	0	0	0	0	0	1	0	0
Lateral	0	0	0	0	0	0	0	0	0	1	0
Aspirated	0	0	0	0	0	0	0	0	0	0	1
Place	0.9	0.85	0.85	0.75	0.75	1	0.85	0.6	0.8	0.85	0
Nasal	0	0	0	0	0	1	1	1	0	0	0

Table A.4: Articulatory feature values for individual phones (y-ao).



phone	y	w	iy	ih	ey	eh	ae	uh	uw	ow	ao
Long	0	0	1	0	1	0.5	0	0.5	1	1	0
Round	0	0	0	0	0	0	0	1	1	1	1
Manner	0.6	0.6	0.4	0.3	0.2	0.2	0.1	0.3	0.4	0.2	0.1
Voice	1	1	1	1	1	1	1	1	1	1	1
High	1	1	1	0.7	0.4	0.4	0.1	0.7	1	0.4	0.1
Back	1	0.1	1	1	1	1	1	0.1	0.1	0.1	0.1
Syllabic	0	0	1	1	1	1	1	1	1	1	1
Retroflex	0	0	0	0	0	0	0	0	0	0	0
Lateral	0	0	0	0	0	0	0	0	0	0	0
Aspirated	0	0	0	0	0	0	0	0	0	0	0
Place	0.7	1	0	0	0	0	0	0	0	0	0
Nasal	0	0	0	0	0	0	0	0	0	0	0

Table A.5: Articulatory feature values for individual phones (ah-oo).



phone	ah	aa	oy	aw	er	ay	ax	axr	ix	oo
Long	0	0.5	1	0	0.5	1	0	0	0	0.5
Round	0	0	0	1	0	0	0	0	0	1
Manner	0.2	0.1	0.1	0.1	0.2	0.1	0.2	0.2	0.3	0.2
Voice	1	1	1	1	1	1	1	1	1	1
High	0.4	0.1	0.1	0.1	0.4	0.1	0.4	0.4	0.7	0.4
Back	0.55	0.1	1.1	0.1	0	0.1	0.55	0	0.55	0.1
Syllabic	1	1	1	1	0	1	1	0	1	1
Retroflex	0	0	0	0	1	0	0	1	0	0
Lateral	0	0	0	0	0	0	0	0	0	0
Aspirated	0	0	0	0	0	0	0	0	0	0
Place	0	0	0	0	0.8	0	0	0.8	0	0
Nasal	0	0	0	0	0	0	0	0	0	0

A.3 EXAMPLE PHONE DISTANCE TABLE

The top row of Table A.6 below lists individual phones, preceded by their numerical identifier. In the columns below this top row each individual column has rows of (1) a phone identifier number, (2) a phone name, and (3) a distance for that row's phone from the column's phone, which is the phone listed in the top row heading the column. Thus each phone in the top row below the heading row has a phonetic distance of 0 from itself. The distance numbers are a figure of merit, with larger numbers meaning greater distance.

Table A.6: Example listings from the phone distance table. The closest phones are *0* distance apart. The artificial *padding* phones (+,_) are the most distant phones.

0 p	1 b	2 t	3 d	4 k	5 g
0 p 0	1 b 0	2 t 0	3 d 0	4 k 0	5 g 0
2 t 6	3 d 6	0 p 6	1 b 6	5 g 10	4 k 10
1 b 10	0 p 10	6 ch 8	7 jh 8	2 t 10	3 d 10
8 f 12	9 v 12	12 s 9	13 z 9	6 ch 11	7 jh 11
10 th 13	11 dh 13	4 k 10	5 g 10	14 sh 16	1 b 16
6 ch 15	7 jh 15	3 d 10	2 t 10	0 p 16	15 zh 16
4 k 16	2 t 16	10 th 12	11 dh 12	12 s 20	13 z 20
3 d 16	5 g 16	8 f 13	15 zh 13	3 d 20	23 w 20
12 s 16	13 z 16	14 sh 13	9 v 13	7 jh 21	2 t 20
14 sh 20	23 w 20	1 b 16	0 p 16	10 th 22	6 ch 21
9 v 22	15 zh 20	7 jh 19	6 ch 19	8 f 24	11 dh 22
11 dh 24	8 f 22	13 z 20	12 s 20	1 b 26	9 v 24
7 jh 25	10 th 24	5 g 20	4 k 20	15 zh 26	22 y 24
13 z 26	6 ch 25	11 dh 22	10 th 22	13 z 30	14 sh 26
5 g 26	4 k 26	15 zh 24	14 sh 24	23 w 30	0 p 26
23 w 30	12 s 26	9 v 24	8 f 24	11 dh 32	12 s 30
15 zh 30	14 sh 30	22 y 36	22 y 26	9 v 34	10 th 32
22 y 42	22 y 32	23 w 36	23 w 26	22 y 34	8 f 34
20 l 46	20 l 36	20 l 40	20 l 30	19 r 48	19 r 38
19 r 48	19 r 38	19 r 42	19 r 31	20 l 50	20 l 40
40 axr 68	37 er 58	40 axr 62	40 axr 52	37 er 68	37 er 58
37 er 68	40 axr 58	37 er 62	37 er 52	40 axr 68	40 axr 58
16 m 70	16 m 60	17 n 70	17 n 60	24 iy 69	30 uw 59
17 n 76	17 n 66	16 m 76	16 m 66	30 uw 69	24 iy 59
24 iy 85	24 iy 75	24 iy 79	24 iy 69	18 ng 70	18 ng 60
30 uw 85	30 uw 75	30 uw 79	30 uw 69	26 ey 74	26 ey 64
18 ng 86	18 ng 76	18 ng 80	18 ng 70	44 oo 74	25 ih 64
41 ix 90	41 ix 80	41 ix 84	41 ix 74	25 ih 74	29 uh 64
29 uh 90	29 uh 80	29 uh 84	29 uh 74	29 uh 74	41 ix 64
44 oo 90	26 ey 80	44 oo 84	26 ey 74	41 ix 74	38 ay 66
26 ey 90	25 ih 80	26 ey 84	25 ih 74	21 hh 75	31 ow 66
25 ih 90	31 ow 82	25 ih 84	31 ow 76	31 ow 76	35 oy 66
21 hh 91	38 ay 82	21 hh 85	38 ay 76	38 ay 76	33 ah 69
31 ow 92	35 oy 82	31 ow 86	35 oy 76	35 oy 76	27 eh 69
35 oy 92	39 ax 85	35 oy 86	39 ax 79	39 ax 79	36 aw 69
38 ay 92	27 eh 85	38 ay 86	27 eh 79	27 eh 79	39 ax 69
33 ah 95	36 aw 85	33 ah 89	36 aw 79	36 aw 79	17 n 70
36 aw 95	33 ah 85	36 aw 89	33 ah 79	33 ah 79	32 ao 74
39 ax 95	32 ao 90	39 ax 89	32 ao 84	17 n 80	28 ae 74
27 eh 95	34 aa 90	27 eh 89	34 aa 84	34 aa 84	34 aa 74
28 ae 100	28 ae 90	28 ae 94	28 ae 84	32 ao 84	16 m 76
32 ao 100	44 oo 100	32 ao 94	44 oo 94	28 ae 84	44 oo 84
34 aa 100	21 hh 101	34 aa 94	21 hh 95	16 m 86	21 hh 85
43 + 200	43 + 210	43 + 194	43 + 204	43 + 184	43 + 194
42 _ 200	42 _ 210	42 _ 194	42 _ 204	42 _ 184	42 _ 194

Appendix B

SHACER Grammars

B.1 GRAMMAR DESCRIPTION AND FORMAT

For SHACER five out of the ensemble of six speech recognizers that are employed (Chapter 5) use our augmented version of Carnegie Mellon University's Sphinx 2 speech recognizer. To augment Sphinx 2 we added the capability for it to use a grammar based language model, as described in Section 4.3 and Figure 4.6. The grammars used for constrained syllable (Section B.2) and phone recognition (Section B.3) and the grammar for the word/phrase-spotting recognizer (Section B.4) are given below.

Grammars are written in the formalism used by PROFER [83, 81], our robust natural language parser — a formalism that is parallel to that used by Carnegie Mellon University's Phoenix parsing system [166] (i.e. PROFER accepts and compiles the same grammars that Phoenix accepts and compiles).

The following rules along with the symbols described in Figure B.8 provide a basis for constructing the grammar definition files used by PROFER.

Figure B.1: Symbols used in grammar definition files.

Character(s)	Meaning
`[ ]`	Square brackets surround the name of a *net*, which show up as "tags" in the output.
`( )`	Parentheses surround each *rewrite pattern* within a net's definition.
`*`	Designates optionality — the term following the `*` may occur or not occur.
`+`	Designates 1 or more occurrences of the term following the `+`.
`*+` or `+*`	Designates 0 or more occurrences of the term following the `*+` or `+*`.

1. Grammars are defined as patterns that get associated with *net* name-tags. Each

individual *net* must have a separate definition file (i.e., a file whose name stem exactly matches the *net* name and whose extension is *.gra*). These points are illustrated in Figure B.2.

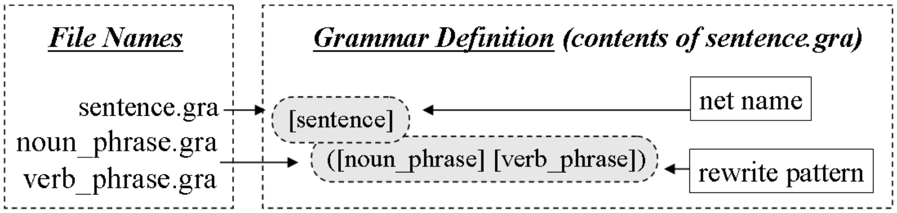


Figure B.2: File name and net name correspondences.

For convenience, multiple nets can be defined in the same file. However, that single file, containing multiple *net* definitions, must at some point be written out into individual files corresponding to the individual *nets* defined within it. This can be done using filization commands that are available in PROFER's application programming interface (API).

2. *Net* names must contain only lower case letters, with no spaces. To separate words within a *net* name underscoring can be used (see the `[noun_phrase]` and `[verb_phrase]` nets in Figure B.2).

3. *Terminals* within rewrite patterns are sequences of all lowercase letters with no underscoring, as shown in Figure B.3 (`john`, `the`, and `boy` are all *terminals*). Note that *terminals* differ from *nets* only in the fact that they are not surrounded by square brackets. Terminals will be matched as strings directly against individual string elements of the input to the parser.

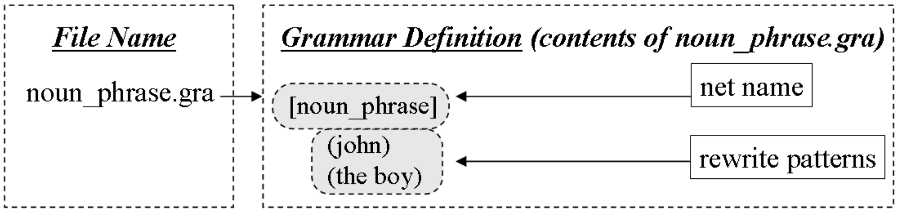


Figure B.3: Terminal rewrite patterns.

4. The top-level *non-terminal* in the definition of a *net* must be a net name, and that name must correspond exactly to the name-stem of the file in which it appears. This

is illustrated in Figure B.2 where `[sentence]` is the top-level *non-terminal* in the file defining the `sentence` *net*, and the file in which this definition occurs is named `sentence.gra` — for the *net* it defines.

5. Subsequent (i.e., non-top-level) *non-terminal* patterns are called *rewrites* and must be all uppercase letters, with no spaces. They may include underscoring to separate words, as illustrated in Figure B.6 by the `OBJECT_PHRASE` *rewrite*.

   In this version of PROFER *rewrites* must be locally defined within each *net* where they appear. Note that *rewrites* do not appear in the output.

6. Non-terminal *rewrites* appearing at the left margin, with no preceding white space begin a block of *rewrite patterns*. Each *rewrite pattern* within the block defines a particular sub-pattern that can be mapped from the input into the block's non-terminal *rewrite*. The pattern list within a block represents a disjunctive list, as illustrated in Figure B.7 — meaning that within a given parse only one of the patterns within the block will actually be used to map the input onto the block's non-terminal *rewrite*.

7. A given non-terminal *rewrite* must not appear in the definition file **below** the block in which it is defined.

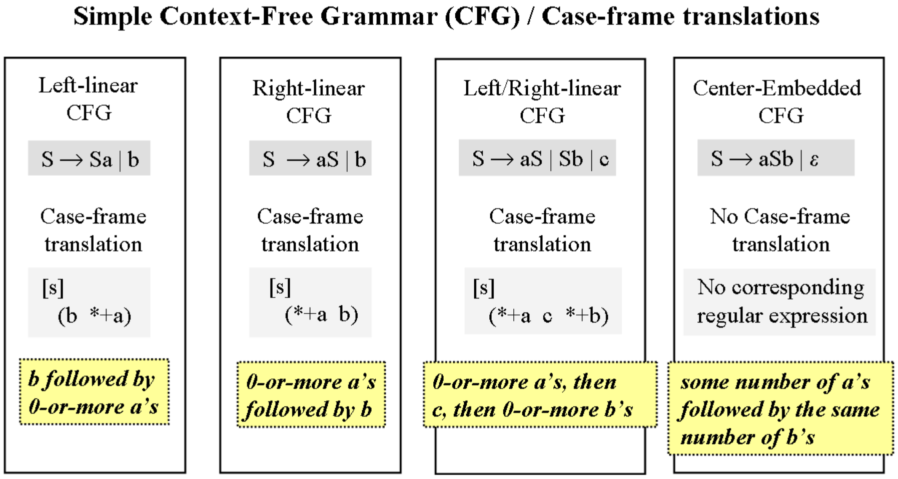

Simple Context-Free Grammar (CFG) / Case-frame translations
Left-linear CFG
S → Sa | b
Case-frame translation
[s] (b *+a)
b followed by 0-or-more a's
Right-linear CFG
S → aS | b
Case-frame translation
[s] (*+a b)
0-or-more a's followed by b
Left/Right-linear CFG
S → aS | Sb | c
Case-frame translation
[s] (*+a c *+b)
0-or-more a's, then c, then 0-or-more b's
Center-Embedded CFG
S → aSb | ε
No Case-frame translation
No corresponding regular expression
some number of a's followed by the same number of b's

Figure B.4: Context-Free translations to case-frame style regular expressions.

8. Since not all Context-Free Grammar (CFG) expressions can be translated into the regular expression formalism employed by our case-frame style parser (as shown in

Figure B.4), some restrictions are necessary. These restrictions rule-out the possibility of "center-embedding" (see the right-most block in Figure B.4) for which there is no equivalent regular language expression:

- A *top-level net name* can appear only once (on the first line) of the file which defines it — and cannot appear within the definition of any *net* that appears within its definition.
- Likewise a *non-terminal rewrite* cannot appear within the block of a *rewrite pattern* by which it is defined.

These restrictions are depicted graphically in Figure B.6.

Note that it is possible to define regular grammars that allow for "center-embedding" of *nets* to any **finite** depth by copying the *net* definition and giving it a unique name for each level of self-embedding desired — two examples are given in Figure B.5. In these examples (**se** stands for **(s)**elf-**(e)**mbedded nets, and **ser** stands for **(s)**elf **(e)**mbedded **(r)**ewrites). Both grammars can parse inputs that contain some number of a's followed by a matching number of b's (up to the level of embedding defined, which in both of these cases is four deep).

```
EXAMPLE: nets              |        EXAMPLE: rewrites
[se]                       |        [ser]
  (a [se_one] b)           |          (a SE_ONE b)
  (a b)                    |          (a b)
[se_one]                   |        SE_ONE
  (a [se_two] b)           |          (a SE_TWO b)
  (a b)                    |          (a b)
[se_two]                   |        SE_TWO
  (a [se_three] b)         |          (a SE_THREE b)
  (a b)                    |          (a b)
[se_three]                 |        SE_THREE
  (a b)                    |          (a b)

INPUT:                     |        INPUT:
  a c a b d e b            |          a c a b d e b
PARSE:                     |        PARSE:
  se:[a,se_one:[a,b],b]    |          ser:[a,a,b,b]
```

Figure B.5: Example of discreet center-self-embedding in regular-expression format.

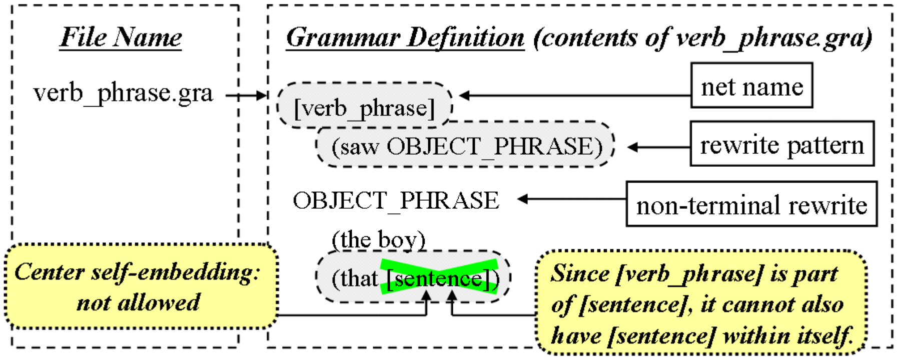

File Name
verb_phrase.gra
Grammar Definition (contents of verb_phrase.gra)
[verb_phrase]
(saw OBJECT_PHRASE)
OBJECT_PHRASE
(the boy)
(that [sentence])
net name
rewrite pattern
non-terminal rewrite
Center self-embedding: not allowed
Since [verb_phrase] is part of [sentence], it cannot also have [sentence] within itself.

Figure B.6: Self-embedding restrictions.

9. Each *rewrite pattern* within a *rewrite* block must be preceded by white space on the line on which it occurs (see Figure B.7).

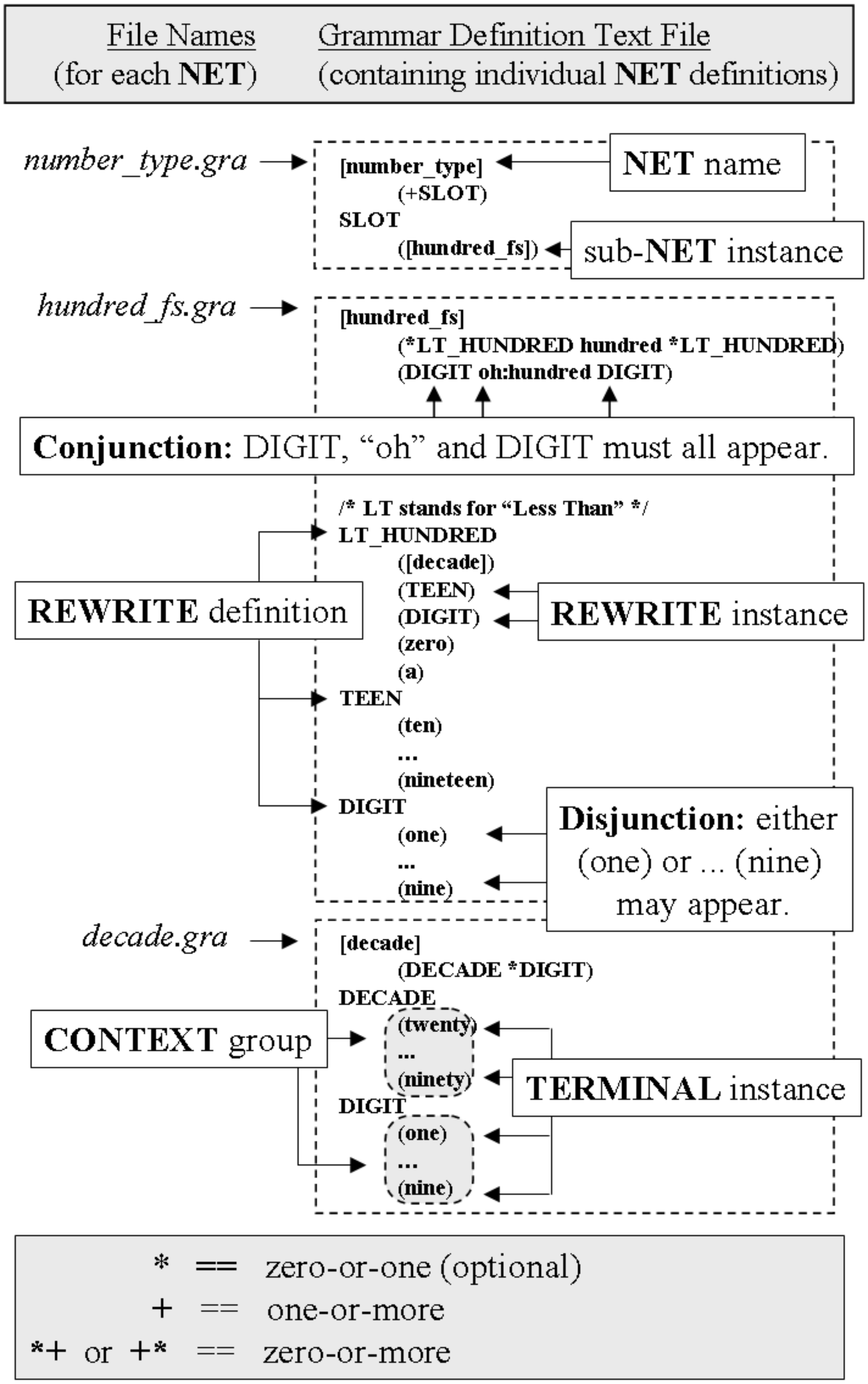
10. Elements within a particular *rewrite pattern* represent a *conjunctive* list; that is, the elements defined in the *rewrite pattern* must be seen sequentially in the input for there to be a valid mapping to the pattern, as shown in Figure B.7. This means that PROFER will not partially recognize an individual pattern. Either a pattern is discovered in the input or it is not. However, elements within a pattern can be marked as optional (`*`), one-or-more (`+`) or zero-or-more (`*+` or `+*`), as described in the symbol table shown in Figure B.8 and illustrated in the *ids.gra* of Figure B.7.

11. Aspects of conjunction and disjunction can be combined to allow for partial parses of the input. Conceptually this amounts to defining *slots* within *frames*, which is the basic *case-frame* architecture of the PHOENIX system.

    PROFER accomplishes the same thing by using conventions to define *slot* or *frame* associations, as shown in the following example:

```
[order]
  (+SLOT)

SLOT
  ([pizza])
  ([drink])
  ([salad])
```

This example defines a *net* called `order` which is composed of any combination of `pizza`, `drink` and `salad` `SLOT`s. An `order` must have at least one of its slots filled; that is, it must include at least one of `pizza`, `drink` or `salad`; but, an order could also include more than one `pizza`, more than one `drink`, more than one `salad` . . . or any combination of such multiples. Thus, if the spoken order is for a `pizza`, a `drink` and a `salad`, but the `drink` portion of the input is garbled, then a partial parse including the portions that fill the `pizza` and `salad` *slots* can still be returned.



```
[number_type]
        (+SLOT)
SLOT
        ([hundred_fs])
```

```
[hundred_fs]
        (*LT_HUNDRED hundred *LT_HUNDRED)
        (DIGIT oh:hundred DIGIT)

/* LT stands for “Less Than” */
LT_HUNDRED
        ([decade])
        (TEEN)
        (DIGIT)
        (zero)
        (a)
TEEN
        (ten)
        ...
        (nineteen)
DIGIT
        (one)
        ...
        (nine)
```

```
[decade]
        (DECADE *DIGIT)
DECADE
        (twenty)
        ...
        (ninety)
DIGIT
        (one)
        ...
        (nine)
```

Figure B.7: Basic formatting for grammar definition files.

B.2 CONSTRAINED SYLLABARY GRAMMAR

This grammar is used for the constrained syllable-sequence phone ensemble recognizer. The grammar was constructed from a syllabified version of the Carnegie Mellon University (CMU) Dictionary version 4.0, which has syllabifications for over 100,000 words. This dictionary was processed to extract lists of first, middle and last word-position syllables as well as single syllable words whose syllables fit in the first-last position syllable category. The grammar specifies the order in which these syllable categories can be combined.

B.3 CONSTRAINED PHONE GRAMMAR

This grammar is used for the constrained phone-sequence phone ensemble recognizer. The grammar was constructed from an existing phone grammar used in [71]. Translated to an appropriate grammar format and used here with permission.

B.4 WORD PHRASE GRAMMAR

The Word/Phrase grammar is the obverse of the MNVR grammar described in Section 4.3.3. That is, rather than primarily recognizing grammar defined carrier-phrase words and occasionally dropping into the secondary syllabic grammar for out-of-vocabulary terms, it does the opposite — it primarily recognizes syllabic terms (e.g. this is shown in the grammar below as being terms in the <*syllabary*> embedded grammar) until it finds a high confidence recognition of an enrolled term (e.g. this is shown in the grammar below as being terms in the <*place_holder_word_phrase_name*> grammar). Initially the *word_phrase_name* grammar, whose only terminal is the <*place_holder_word_phrase_name*>, is always empty, but its filled at start-up time from any designated and available file-based accumulators like those described in Appendix C.

B.4.1 Word Phrase

```
[word_phrase]
    (+[term])

[term]
    ([syl_terms])
    ([word_phrase_name])
    ([sem_wp])
```

```
[syl_terms]
    (<syllabary>)

[word_phrase_name]
    (<place_holder_word_phrase_name>)

[sem_wp]
    (sem_taskline)
    (sem_line)
    (sem_milestone)
```

B.4.2 Syllabary

```
[syllabary]
    ([first_last_syllable])
    ([first_last_syllable] [first_last_syllable])
    ([first_syllable] [last_syllable])
    ([first_syllable] [last_syllable] [first_last_syllable])
    ([first_last_syllable] [first_syllable] [last_syllable])
    ([first_syllable] [middle_syllable] [last_syllable])

[first_last_syllable]
    (z_aa)
    (z_aa_b_s_t)
    (z_aa_d)
    (z_aa_d_z)
    \ldots (first_last_syllable contains 9606 syllabic terminals)
    (z_zh_ey_k)
    (z_zh_iy_l)
    (z_zh_w_aa)
    (z_zh_w_aa_n)

[first_syllable]
    (z_aa)
    (z_aa_b)
    (z_aa_b_f)
    (z_aa_b_s)
    \ldots (first_syllable contains 17259 syllabic terminals)
    (z_zh_w_aa)
    (z_zh_w_aa_n)
    (z_zh_w_aa_r)

[middle_syllable]
    (z_aa)
    (z_aa_b)
    (z_aa_b_s)
    (z_aa_ch)
    \ldots (middle_syllable contains 2988 syllabic terminals)
    (z_zh_aa_r)
    (z_zh_ax_k)
    (z_zh_er)
    (z_zh_w_aa_z)

[last_syllable]
    (z_aa)
    (z_aa_b)
    (z_aa_ch)
    (z_aa_d)
    \ldots (middle_syllable contains 3775 syllabic terminals)
    (z_zh_ih_n)
    (z_zh_iy_n)
    (z_zh_w_aa)
```

B.5 SHACER Stop-List

A *stop-list* typically contains closed class words like articles, prepositions, pronouns and auxiliary verbs that tend to occur with equal frequency in most documents, making them not useful as representative words for a particular document.

- a
- aboard
- about
- above
- across
- after
- against
- all
- along
- alongside
- although
- am
- amid
- amidst
- among
- amongst
- an
- and
- another
- anti
- any
- anybody
- anyone
- anything
- are
- around
- as
- astride
- at
- aught
- bar
- barring
- be
- because
- been
- before
- behind
- being
- below
- beneath
- beside
- besides
- between
- beyond
- both
- but
- by
- can
- circa
- concerning
- considering
- could
- despite
- down
- during
- each
- either
- enough
- everybody
- everyone
- except
- excepting
- excluding
- few
- fewer
- following
- for
- from
- have
- he
- her
- hers
- herself
- him
- himself
- his
- hisself
- i
- idem
- if
- ilk
- in
- including
- inside
- into
- is
- it
- its
- itself
- like
- many
- may
- me
- might
- mine
- minus
- more
- most
- must
- myself
- naught
- near
- neither
- nobody
- none
- nor
- nothing
- notwithstanding
- of
- off
- on
- oneself
- onto
- opposite
- or
- other
- otherwise
- ought
- our
- ourself
- ourselves
- outside
- over
- own
- past
- pending
- per
- plus
- regarding
- round
- save
- self
- several
- shall
- she
- should
- since
- so
- some
- somebody
- someone
- something
- somewhat
- such
- suchlike
- sundry
- than
- that
- the
- thee
- theirs
- them
- themselves
- there
- they
- thine
- this
- thou
- though
- through
- throughout
- thyself
- till
- to
- tother
- toward
- towards
- twain
- under
- underneath
- unless
- unlike
- until
- up
- upon
- us
- various
- versus
- via
- vis-a-vis
- was
- we
- were
- what
- whatall
- whatever
- whatsoever
- when
- whereas
- wherewith
- wherewithal
- which
- whichever
- whichsoever
- while
- who
- whoever
- whom
- whomever
- whomso
- whomsoever
- whose
- whosoever
- will
- with
- within
- without
- worth
- would
- ye

Figure B.8: SHACER's stop list.

Appendix C

Learning Accumulator Files

The file contents listed below are also explained briefly in Section 7.4.

C.1 SPEECH AND HANDWRITING DICTIONARY ADDITIONS

The accumulator file contents below are from the *sphinx_add_to_dict.log* file resulting from the processing of the fourth meeting in the development series, which is referred to as the *G* series. Note that processing of this fourth meeting depends on the processing of the third meeting, which in turn depended on those before it. Thus the file list below is effectively an accumulation from all four previous meetings. The lines are listed in groups of five, each starting with the system's best scoring orthographic hypothesis of the word to be newly enrolled in the Word/Phrase-Spotting recognizer's (WPSR's) dictionary and grammar. The second terms in each line form a list of alternate spellings, which are informative but not used as part of the new enrollment. The third terms in each line list alternative pronunciations for the new term. The process of choosing the best spelling and pronunciations is described in Chapter 5.

SHACER's WPSR is an implementation of CMU's Sphinx 2 speech recognizer — which we have augmented to use rule-based language models defined by grammars like those described in Appendix B. Thus the WPSR, as does Sphinx 2, uses acoustic tri-phones as the basis for modeling spoken linguistics. For optimal efficiency the recognizer is designed to determine the minimal set of tri-phones necessary to support recognition given its dictionary. To do this the recognizer looks at both all word-internal tri-phones and also all word-transitional tri-phones. Word transitional tri-phones are determined by taking each dictionary word and first pairing it with all dictionary words in a following position (to create a so-called right-context table of tri-phone transitions) and then pairing it with all dictionary words in a preceding position (to create a so-called left-context table of tri-phone transitions). This guarantees that all possible tri-phone contexts that the system will need for recognition are available from the recognizer’s database of Gaussian mixture models and their model weights.

Dynamically adding new words to the WPSR dictionary sometimes requires an accompanying dynamic expansion of these tri-phone context databases, otherwise the system fails. As designed Sphinx 2 only performs the construction of its tri-phone database context tables when its language model is initially read in. We modified Sphinx 2 to support dynamic run-time expansion of its tri-phone context databases. During run-time this ability to dynamically expand the tri-phone context databases is not part of the recognition process thread, so it is possible for recognition to restart while the tri-phone context databases are still being rebuilt. To keep this from happening we designed a method of first caching the additions to be made and then adding them within a routine that blocked further recognition until the expansion process was complete.

Currently we are not clustering and refining pronunciations for the same word as they are entered, as is described by Yu [175] for their experiments in recognizing embodied intentions. So, for an entry like *Joe Browning* there are five separate instances (listed as five separate groups of *Joe Browning* entries in the file below) during the first four of the *G* series meetings in which redundant handwriting and speech information was combined with high enough confidence to trigger an enrollment. Given these five enrollment groups there are about a dozen separate pronunciations for *Joe Browning* that will be active for recognition during the fifth *G* series meeting. All of these separate pronunciation alternatives are added to the Word/Phrase-Spotter’s dictionary and considered with equal likelihood at present. Part of our future work will be clustering and refining pronunciation alternatives. Given the current relatively small size of the enrolled dictionary (numbering in the dozens of terms) and our focus on developing and testing a proof-of-concept learning system, such pronunciation refinements have not been a priority for the purposes of this thesis dissertation.

```
log_entry(sphinx_add_to_dict,sphinx_add_to_dict(
  ’Joe_Browning’,’Joe_Browning’,’jh ow b r aw n ih ng’,[])
  ’Joe_Browning’,’Joe_Browning’,’jh ow b r aw n r ih ng’,[])
  ’Joe_Browning’,’Joe_Browning’,’jh ow b r aw n d ih ng’,[])
  ’Joe_Browning’,’Joe_Browning’,’jh ow b r aw n th ih ng’,[])
  ’Joe_Browning’,’Joe_Browning’,’jh ow b r aw n r ih ng z’,[])
  ’Arrive’,’Arrive’,’er ay v’,[])
  ’Arrive’,’Arrives’,’er ay v z’,[])
  ’Arrive’,’Arrives’,’t uw er ay v z’,[])
```

```
'Arrive','ARRIVE','t uw er ay v',[])
'Arrive','ARRIVE','er aa ay v',[])
'Buy_Computer','Buy_Computer','b ay k ah m p y uw t er',[])
'Buy_Computer','Buy_Computer','k ah m p y uw t er',[])
'Buy_Computer','BuyComputer','ao v k ah m p ch y uw uh r',[])
'Buy_Computer','Buy_Computer','d ih m p ch y uw hh iy uh r er',[])
'Buy_Computer','Buy_Computer','ah v k ah m p ch y uw uh r',[])
office,office,'ao f ah s',[])
office,office,'ao f ah s ow',[])
office,office,'ao f ah s ah m',[])
office,office,'ow ao f ah s',[])
office,office,'ao l f ah s',[])
proposed,proposed,'p r ah p ow z d',[])
proposed,proposed,'p r ah p ow z',[])
proposed,proposed,'p r ah p ow z d t uw',[])
proposed,proposed,'ah p r v p ow uw z t',[])
proposed,trop_used,'p r v p hh aa l uw z t',[])
equipment,equipment,'ih k w ih p m ah n t',[])
equipment,equipment,'ih k w ih p m ah n t jh iy',[])
equipment,equipment,'ih k w ih p m ah n t d uw',[])
equipment,equipment,'ih k w ih p m ah n t r ih p',[])
equipment,equipment_t,'ih k w ih p m ah n t t ey p',[])
'Available','Available','ah v ey l ah b ah l',[])
'Available','Available','ah v ey l ah b ah l t uw',[])
'Joe_Browning','Joe_Browning','jh ow b r aw n ih ng',[])
'Joe_Browning','Joe_Browning','jh uw b r aw n ih ng',[])
'Joe_Browning','Joe_Browning','jh y ow b r aw n ih ng',[])
'Joe_Browning','Joe_Browning','jh uw l b r aw n ih ng',[])
'Joe_Browning','Joe_Browning','jh uw b r aw m ih ng',[])
'Fred_Green','Fred_Green','f r eh d g r iy n',[])
'Fred_Green','Fred_Green','th r eh d g r iy n',[])
'Fred_Green','Fred_Green','f r eh d z g r iy n',[])
'Fred_Green','Fred_Green','f r eh d k r iy n',[])
'Fred_Green','Fred_Green','f er eh d g r iy n',[])
'Cindy_Black','Cindy_Black','s ih n d iy b l ae k',[])
'Cindy_Black','Cindy_Black','s ih n d iy z b l ae k',[])
'Cindy_Black','Cindy_Black','s ih n p iy d b ah l ae ae k',[])
'Cindy_Black','Cindy_Black','s ih n d iy d w ay v k',[])
'Cindy_Black','Cindy_Black','s ih n p iy d w ay v k',[])
'Arrive','Arrive','er ay v',[])
'Arrive','ARRIVE','er aa ay v s',[])
'Arrive','ARRIVE','er ao ay v s',[])
'Arrive','ARRIVE','w er ay v s',[])
'Arrive','ARRIVE','w er ay v z',[])
'Buy_Computer','Buy_Computer','b ay k ah m p y uw t er',[])
'Buy_Computer','BUY_COMPUTER','w ay k ah m p y uw r',[])
'Buy_Computer','BUY_COMPUTER','w ay k ah m p y er',[])
'Buy_Computer','BUY_COMPUTER','w ay g ah m p y uw r',[])
'Buy_Computer','BUY_COMPUTER','w ay g ah m p y er',[])
```

```
’Recommend’,’Recommend’,’r eh k ah m eh n d’,[])
’Recommend’,’Recommend’,’r eh k ah m eh n d d iy’,[])
’Recommend’,’Recommend’,’r eh k ah m eh n d d uw z’,[])
’Office’,’Office’,’ao f ah s’,[])
’Office’,’OFFICE’,’ao f ah’,[])
’Office’,’OFFICE’,’f ah z’,[])
’Office’,’OFFICE’,’ao dh ah’,[])
’Office’,’OFFICE’,’ao l f ah’,[])
’Recommend’,’Recommend’,’r eh k ah m eh n d’,[])
’Recommend’,’RECOMMEND’,’r eh k m ah n b’,[])
’Recommend’,’RECOMMEND’,’r eh k w eh n b’,[])
’Recommend’,’RECOMMEND’,’r ah k w ih n b’,[])
’Recommend’,’RECOMMEND’,’r eh k w uh n b’,[])
joe_browning,joe_browning,’jh ow b r aw n ih ng’,[])
joe_browning,joe_browning,’jh ow b b r aw n iy’,[])
joe_browning,joe_browning,’jh r ow p b r aw m ih’,[])
joe_browning,joe_browning,’ch uw w b r aw n ih’,[])
joe_browning,joe_browning,’ch uw w b r aw n ey’,[])
’Fred_Green’,’Fred_Green’,’f r eh d g r iy n’,[])
’Fred_Green’,’Fred_Green’,’f r eh d k r iy n’,[])
’Fred_Green’,’Fred_Green’,’f r ih t g r iy n’,[])
’Fred_Green’,’Fred_Green’,’f r ih d g r iy n’,[])
’Fred_Green’,’Fred_Green’,’f r ih d k r iy n’,[])
joe_browning,joe_browning,’jh ow b r aw n ih ng’,[])
joe_browning,joe_browning,’jh ow b r aw n ih ng r ay’,[])
joe_browning,joe_browning,’jh ow b r aw n iy ng r ay’,[])
joe_browning,joe_browning,’jh ow b r aw n ey ng r ay’,[])
joe_browning,joe_browning,’jh ow b r aw n iy n r ay’,[])
’Cindy_Black’,’Cindy_Black’,’s ih n d iy b l ae k’,[])
’Cindy_Black’,’Cindy_Black’,’s ih n d iy b l ay ae k’,[])
’Cindy_Black’,’Cindy_Black’,’s ih n y uw b w ae k’,[])
’Cindy_Black’,’Cindy_Black’,’s ih n y uw b l ay k’,[])
’Cindy_Black’,’Cindy_Black’,’s ih n y iy b w ay k’,[])
fred_green,fred_green,’f r eh d g r iy n’,[])
fred_green,fred_green,’f r ey d g r iy n’,[])
fred_green,fred_green,’f r eh d g r iy ng’,[])
fred_green,fred_green,’f r eh ih d g r iy n’,[])
fred_green,fred_green,’th r eh d g r iy n’,[])
’Arrive’,’Arrive’,’er ay v’,[])
’Arrive’,’Arrive’,’er ay v iy’,[])
’Arrive’,’Arrive’,’er ay v d’,[])
’Arrive’,’Arrive’,’er w ay v y’,[])
’Arrive’,’Arrive’,’er w ay v dh’,[])
’Buy_Computer’,’Buy_Computer’,’b ay k ah m p y uw t er’,[])
’Buy_Computer’,’Buy_Computer’,’hh w ay k ah m p y uw t er’,[])
’Buy_Computer’,’Buy_Computer’,’ay k ih m p y uw t uw l’,[])
’Buy_Computer’,’Buy_Computer’,’ay k ih m p y uw dh uw l’,[])
’Buy_Computer’,’Buy_Computer’,’ay k ih m p y uw d uw l’,[])
’Recommend’,’Recommend’,’r eh k ah m eh n d’,[])
```

```
'Recommend','RECOMMEND','r w eh k uw m ih n d',[])
'Recommend','RECOMMEND','r w ih k ah m eh n d dh',[])
'Recommend','RECOMMEND','r w ih k ah m ih n d dh',[])
'Recommend','RECOMMEND','r w ih k ah m ah n d dh',[])
'Recommend','Recommend','r eh k ah m eh n d',[])
'Recommend','RECOMMEND','k uw ah n d',[])
'Recommend','RECOMMEND','t k uh ih n d',[])
'Recommend','RECOMMEND','k uw ih n b',[])
'Recommend','RECOMMEND','t g uw ah n d',[])
office,office,'ao f ah s',[])
office,office,'ao f ah s p iy',[])
office,office,'ao f ih s',[])
office,office,'ao f ih z',[])
office,office,'ao f ah z',[])
'Avail','Avail','ah v ey l',[])
'Avail','Avail','ih b eh l',[])
'Avail','Avail','ih t d ae ow l',[])
'Avail','Avail','ih t b ae ow l',[])
'AVAILABLE','AVAILABLE','ah v ey l ah b ah l',[])
'AVAILABLE','AVAILABLE','ah v b iy l p ah l',[])
'AVAILABLE','AVAILABLE','ah v dh ih ow l b l',[])
'AVAILABLE','AVAILABLE','ah v dh eh l b l',[])
'AVAILABLE','AVAILABLE','ah v dh ih l b l',[])
joe_browning,joe_browning,'jh ow b r aw n ih ng',[])
joe_browning,joe_browning,'jh uw b r aw n ih ng',[])
joe_browning,joe_browning,'jh ow b r ah n ih ng',[])
joe_browning,joe_browning,'jh uw b r ah n ih ng',[])
joe_browning,joe_browning,'y uw l p r ah n ih ng',[])
cindy_black,cindy_black,'s ih n d iy b l ae k',[])
cindy_black,cindy_black,'s ih n d iy b w ae k',[])
cindy_black,cindy_black,'s eh n d iy b w ae k',[])
cindy_black,cindy_black,'s ih n jh iy b w ay k',[])
cindy_black,cindy_black,'s ih n d iy p w ay ae k',[])
fred_green,fred_green,'f r eh d g r iy n',[])
fred_green,fred_green,'v r eh d g r iy n',[])
fred_green,fred_green,'v r eh p g r iy n',[])
fred_green,fred_green,'v r ey d g r iy n',[])
fred_green,fred_green,'v r ey p g r iy n',[])
'Buy_Computer','Buy_Computer','b ay k ah m p y uw t er',[])
'Buy_Computer','Buy_Computer','k ah m p y uw t er',[])
'Buy_Computer','Buy_computers','k ah m p y uw t er z',[])
'Buy_Computer','Buy_Computer','k uw m p y uw t r',[])
'Buy_Computer','Buy_Computer','k uw m p y uw t uh r',[])
'Office','Office','ao f ah s',[])
'Office','OFFICE','ao l f ah s',[])
'Office','OFFICE','ao l f ah',[])
'Office','OFFICE','ao l th ah s',[])
'Office','OFFICE','aa l f ah s',[])
'Recommend','Recommend','r eh k ah m eh n d',[])
```

```
  'Recommend','RECOMMEND','r eh k ah m ih n d ih',[])
  'Recommend','RECOMMEND','r eh k ah m ih n d ey',[])
  'Recommend','RECOMMEND','r eh k ah m ih n d dh',[])
  'Recommend','RECOMMEND','er eh k er m ih n d ey',[])
)
```

C.2 ABBREVIATION TABLE INFORMATION

As abbreviations are learned (Section 6.4) each abbreviation/expansion pair is enrolled in an abbreviation lookup table. The list below is a cumulative log of those enrollments for the first four meetings of the *G* series. This table aids in subsequent abbreviation recognition, for example, by allowing lookups of the handwriting recognizer's letter sequence hypotheses so that their expansions can be compared to the speech transcript (as described in Section 5.3.2) for immediate association of known handwritten-abbreviation with their transcribed redundant speech. When the list below is enrolled at the start of the *G5* meeting, exact duplicates will be filtered out, but orthographic alternatives will be included.

```
JB | Joe Browning
JB | joe browning
FG | Fred Green
JB | joe browning
CB | Cindy Black
FG | fred green
Avail | AVAILABLE
JB | joe browning
CB | cindy black
FG | fred green
```

C.3 MIS-RECOGNITION TABLE INFORMATION

The entries listed below from the *sphinx_add_to_reco_hw_nbest_table.log* file provide the input to the master speech recognizer's prefix/suffix mis-recognition reinforcement table (see Section 7.3), which associates high confidence recognitions with their list of related mis-recognitions and significant affixes. We refer to affixes because mis-recognition in SHACER is often triggered by ink skips, which tend to come at the beginning or ending of terms — thus producing prefix/suffix letter string recognitions. Our intuition in regard to ink-skip affix mis-recognitions is that the aspects of a user's handwriting, which cause or are related to the ink skip, may remain consistent for that handwriter — so the same affix mis-recognitions may occur over and over. When affix misrecognitions are discovered, which are very highly correlated to a previous recognition, that consistent recognition error can be transformed into its correct interpretation. The entries in this file are made each time there is a high confidence integration of redundant handwriting and speech.

Each entry line below is added to the log file in parallel to the enrollment of a new word in the Word/Phrase-Spotter recognizer. Each entry's first parameter is the orthographic representation of the new term being enrolled, which serves as a string-key for the hash-table structure that is used to hold this lookup table's information. The values associated with each hashed string-key are those listed in the entry as the second parameter. This list includes all handwriting recognizer alternate outputs for the enrolled term. The list is used subsequently during handwriting and speech integration to identify handwriting recognizer outputs that have been seen before and mis-recognized in the same form. When that mis-recognition is substantially unique (i.e., above a threshold value of association with a particular new word) then the system regards that the new word as having been the actual handwriting entry. This can account for succeeding in label recognitions that otherwise could not be correctly recognized by the system (e.g. Section 7.6.1 and 7.4).

```
log_entry(sphinx_add_to_reco_hw_nbest_table,sphinx_add_to_reco_hw_nbest_table(
    'Joe_Browning',['For Browning','Joe Browning','Foe Browning','Toe
Browning','Hoe Browning','Joc Browning','Jos Browning','Fore Browning','Jol
Browning','foe Browning'])

    'Arrive',['ARUM','ARRIVE','Arrive,','ARRUDA','Arrives','ARRIVER','ARRIVES'])

    'Buy_Computer',['Buy Computer','BuyComputer','Busy
Computer','Buycomputer','Buoy Computer','Buycomputw','Bury
Computer','BuyComputw','Briny Computer','BuyCompufer'])

    office,['of fire','of fie','of fine','of free','of Fie','of fice','of
fico','of Grice','of Spice'])

    proposed,['trop used','Proposed','Hop used',proposal,'top
used','propose,','Ho posed','to posed','to posed','HO posed'])

    equipment,['equip meant','equipment t','equip merit','Equipment t','equip
ment','equipmen t','equip meat','eyuipmen t','equip mint'])

    'Available',['Avail able','Avail ab he','Avail ab le','Avail ab ie','Avail
ab Ie','Avail ab foe','Avail ab fie','Avail ab toe','Avail ab hoe'])

    'JB',['TB','JOB','JIB','TAB','JAB','Jos','JP','J-B','j-B'])

    'Fred_Green',['i-redesign','Erred Green','l-redesign','Erect
```

```
Green','t-retirees','Eared Green','l-redlines','t-red
Green','i-redlines','l-red Green'])

    'Cindy_Black',['Kindy Black','Cindy Black',industrials,'Lindy
Black',indusial,'iindy Black','Kindly Black'])

    'Arrive',[arrive,'Arvin',avrivc,'Avrivc',arrivi,avrive,avrivi,avriVc,arriva])

    'Buy_Computer',['lay computer','1 uycemputer','1 uycempirter','1
uycempater','1
uyeanputer',luyeanputer,burglarproofed,buyouts,buoyantly,'Honeycombs'])

    'Recommend',['Reoco mm end','Reocommend','Reoc0 mm
end','Reocummend','Rococo mm end','Reocommenol','Revco mm
end','Reocommond','Reoccur mm end'])

    'Office',[office,'off ice','of fico','of fire','of file','of twice','of
fice','of trice','of fide','of fiche'])

    'Recommend',[recommend,'Recommencl','Rocommend','Rccommend','Recommenct',
    recommencl,'Recommenci',recommenci,'Re commend'])

    'JB',['5 B','TB','Js','Jos','J B','j B','5 B','y B','I B'])

    'FG',['I-G','H-G','IT-G','Io-Gs','EG','To-Gs','HG','Io-GL','I-G,'])

    'JB',['Jos','Js','JD','JDs','Job','Jas','Jog',jd])

    'CB',[cos,as,cy,cd,cry,coy,ay,'Coy',cis])

    'FG',['EG','Visor','i-g',tg,'Tier','i-or','Ii-or','Ti-or','ii-or'])

    'Arrive',['ARNE',arrive,'ARNIE','ARCANE','AR Rue','AR Rive','AR RWE','AR
RUNE','AR RNs'])

    'Buy_Computer',['Buy computer','Buycomputer','Buycimputer','Buyamputer',
    'Buyeimputer','Buyamputcr','Bumper','Bumpier','Bumpers','Bumpiest'])

    'Recommend',[recommends,recommend,'1 ecoMMEWD',iecoMMEWD,lecoMMEWD,
    recommender,recommenders,'Recommender','Recommenders'])

    'Recommend',[recommend,recommender,'become ND','become END',recommended,
    'Become ND','Secom END','becomes ND','Become END','1 econmenD'])

    office,['Office','off
ice','oft-ice','off-ice',officc,'of-fid',offici,'oft-id',officC])

    'Avail',[avail,'Artie','Arthur','Arian','Avian',avian,'Arte',
    'Artier','Arties'])
    ’Avail’,[trail,trait,avail,’t-rail’,’Argil’,’Arai,’,’trail.’,
    ’trail,’,’Argil,’])

    ’JB’,[’TB’,’5 B’,’JOB’,’JIB’,’J B’,’j B’,’5 B’,’y B’,’f B’])

    ’CB’,[iB,’SB’,’Cts’,’EB’,eB,cps,cB])

    ’FG’,[’T-G’,’F-G’,’I-O’,’FT-G’,’t-g’,’ET-G’,’l-g’,’i-g’])

    ’Buy_Computer’,[’Buy Computer’,’BuyCoarputer’,’Buy
computer’,’Buyloarputer’,’Buy computers’,’Buyloonputer’,’Buy
Coarputer’,’Buyloaiputer’,’Buy Coanputer’])

    ’Office’,[office,’oft-ice’,’of-fie’,’of-tice’,offici,’of-tie’,
    offie,’of-fisc’,’oft-ie’])

    ’Recommend’,[’Recommencl’,’Recommenced’,’Recommenci’,’Recommenc’])
)
```

C.4 HANDWRITING RECOGNIZER BIAS INFORMATION

As explained in Section 7.4 the lines of the `vocab_nbest.txt` file below populate the handwriting recognizer’s weight biasing mechanism, which boosts more frequently recognized terms to higher positions on the handwriting recognizer’s output alternates list.

The entry lines below are added whenever a label is recognized and assigned during Charter meeting processing, so the entries are for both integrated handwriting/speech labels and also for labels assigned by handwriting recognition alone without the benefit of a high confidence redundancy integration. Each addition increases to the term’s frequency count maintained in the handwriting recognition agent. The effect of these entries in the handwriting recognizer is that, for example, after *JB* has been used and recognized as a label four times in meetings *G1* - *G4* (as reflected in the four file entries below and the four times that *JB* appears in the *int* + *speech* illustrations shown in Sections D.4, D.5, D.6 and D.7) then in meeting *G5* an instance of *JB* occurring in the alternates list resulting from handwriting recognition will receive an appropriately biased score, which is likely to move it up in term of its rank on that list of alternates. This means that frequently used and recognized terms get recognized by the handwriting recognizer more easily over time and usage.

```
Joe Browning | 1
Arrive | 1
Buy Computer | 1
office | 1
proposed | 1
equipment | 1
space | 1
Available | 1
JB | 1
Arrive | 1
Fred Green | 1
Cindy Black | 1
Arrive | 1
lay computer | 1
Buy Computer | 1
Recommend | 1
Office | 1
Recommend | 1
JB | 1
CB | 1
FP | 1
sr | 1
FG | 1
JB | 1
Arrive | 1
CB | 1
FG | 1
Arrive | 1
Buy Computer | 1
Recommend | 1
Recommend | 1
office | 1
office | 1
Avail | 1
Avail | 1
Avail | 1
trail | 1
Avail | 1
Recommend | 1
JB | 1
cb | 1
CB | 1
FG | 1
Buy Computer | 1
Office | 1
Recommend | 1
Recommend | 1
1 K | 1
Z | 1
```

```
KC | 1
avail | 1
avail | 1
```

Appendix D

Development Set Meeting Transcripts

D.1 SPHINX 2 PHONE SET

D.1.1 Noise Phones

There are nine noise phones used in SHACER's versions of the Sphinx 2 speech recognizer.

```
+BREATH+, +CLICK+, +DOOR+,
+LAUGH+, +MIKE+, +NOISE+,
+NONVOCAL_NOISE+, +SOFT_NOISE+, +VOCAL_NOISE+
```

D.1.2 Speech Phones

There are forty speech phones used in the SHACER versions of the Sphinx 2 speech recognizer.

```
AA, AE, AH, AO, AW,
AY, B, CH, D, DH,
EH, ER, EY, F, G,
HH, IH, IY, JH, K,
L, M, N, NG, OW,
OY, P, R, S, SH,
SIL, T, TH, UH, UW,
V, W, Y, Z, ZH
```

D.2 LVCSR LATTICE EXAMPLE

The following is an elided version of a lattice file produced by Speechalyzer (see Section 5.4.1).

```
...
Frames 264
#
Nodes 232 (NODEID WORD STARTFRAME FIRST-ENDFRAME LAST-ENDFRAME)
```

```
0 </s> 264 264 264
1 <sil> 212 236 263
2 ++BREATH++ 212 220 235
3 ARE(2) 204 207 217
4 OR(2) 204 206 217
5 THEIR 179 198 217
6 THEY'RE 179 198 217
7 THERE 179 197 218
8 THERE'RE 178 202 203
9 THEIR 178 198 217
10 THEY'RE 178 198 218
11 THERE 178 197 218
12 OTHER 170 190 219
...
142 I'M 20 33 46
143 IN(2) 20 32 44
144 THE(2) 20 30 33
145 TWO 20 30 34
146 THE(3) 20 29 31
147 TO(3) 20 28 33
148 I 20 26 28
149 COMPUTER 19 68 116
150 COMPUTE 19 64 77
151 DUMPS 19 50 55
152 DUMP 19 47 57
153 DIDN'T(4) 19 44 48
154 AM 19 39 44
155 COME 19 32 48
156 TWO 19 31 32
...
218 PIPE 1 19 28
219 POINT 1 19 28
220 BUTT 1 18 28
221 PART 1 18 28
222 BEEN(2) 1 17 20
223 BUT 1 16 30
224 WHY(2) 1 15 17
225 BUY 1 13 18
226 BY 1 13 19
227 BYE 1 13 19
228 PER 1 13 17
229 BOY 1 12 16
230 <sil> 1 2 12
231 <s> 0 0 0
#
Initial 231
Final 0
#
BestSegAscr 0 (NODEID ENDFRAME ASCORE)
```

```
#
Edges (FROM-NODEID TO-NODEID ASCORE)
1 0 -166492
3 1 -150221
3 2 -150221
4 1 -150221
4 2 -150221
5 1 -449761
5 2 -449761
5 3 -340703
5 4 -340703
...
230 198 -38822
230 199 -38822
230 200 -38822
230 201 -38822
230 202 -38822
230 203 -38822
230 204 -38822
230 205 -38822
230 206 -38822
231 207 0
231 208 0
...
231 226 0
231 227 0
231 228 0
231 229 0
231 230 0
End
```

D.3 LVCSR TRANSCRIPT EXAMPLE

The following is a formatted version of an individual utterance's transcript file extracted from the larger Speechalyzer output file.

```
   UTTERANCE: (
a.   01050427Z005351,
b.   2005-05-24T14:42:37.516,
c.   2005-05-24T14:42:40.059,
d.   BUT COMPUTER AND OTHER,(
     (
d.     2005-05-24T14:42:37.516,
e.     2005-05-24T14:42:37.716,
f.     BUT),
     (
d.     2005-05-24T14:42:37.725,
e.     2005-05-24T14:42:38.535,
f.     COMPUTER),
     (
d.     2005-05-24T14:42:38.544,
e.     2005-05-24T14:42:39.011,
f.     AND),
     (
d.     2005-05-24T14:42:39.020,
e.     2005-05-24T14:42:39.087,
f.     <sil>),
     (
d.     2005-05-24T14:42:39.097,
e.     2005-05-24T14:42:39.525,
f.     OTHER),
     (
d.     2005-05-24T14:42:39.535,
e.     2005-05-24T14:42:40.020,
f.     <sil>),
     (
d.     2005-05-24T14:42:40.030,
e.     2005-05-24T14:42:40.030,
f.     </s>)))
```

D.4 G1 MEETING

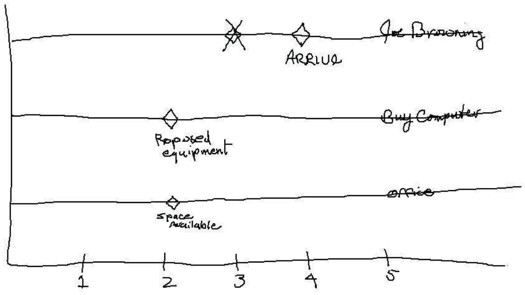


Figure D.1: Ink for the G1 Meeting.

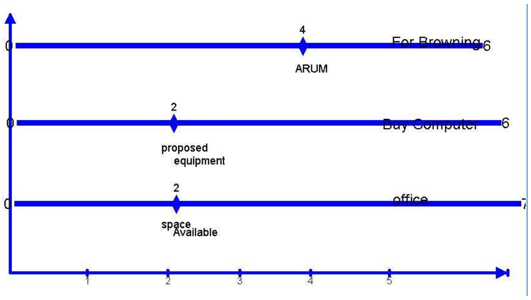


Figure D.2: Phase 1 (ink-only) processing of the G1 Meeting.

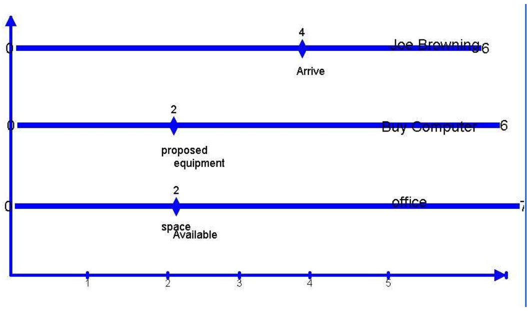


Figure D.3: Phase 2 (ink + speech) processing of the G1 Meeting.

D.5 G2 MEETING

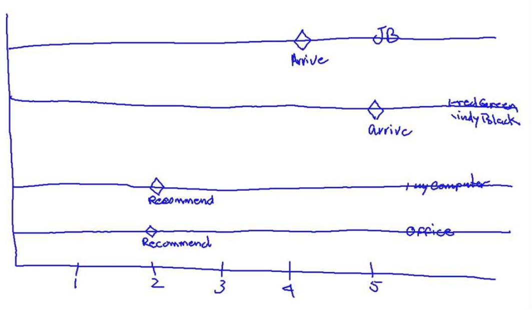


Figure D.4: Ink for the g2 Meeting.

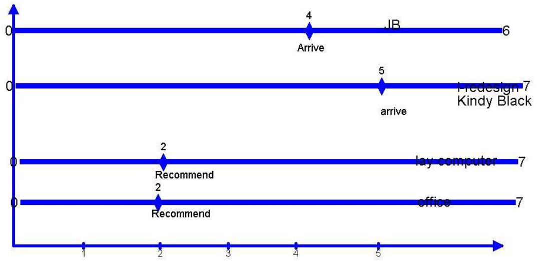


Figure D.5: Phase 1 (ink-only) processing of the g2 Meeting.

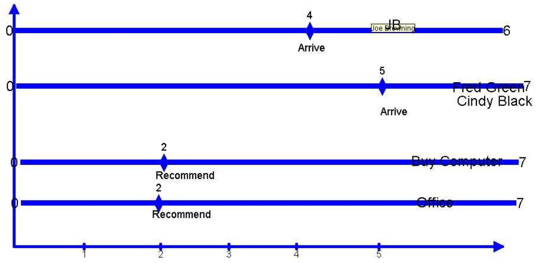


Figure D.6: Phase 2 (ink + speech) processing of the g2 Meeting.

D.6 G3 MEETING

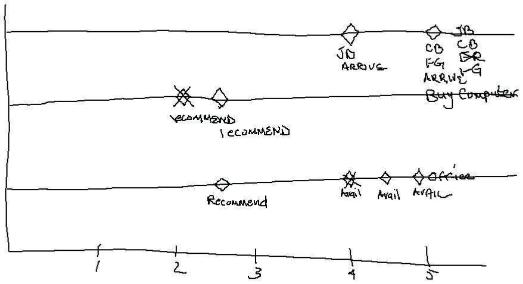


Figure D.7: Ink for the g3 Meeting.

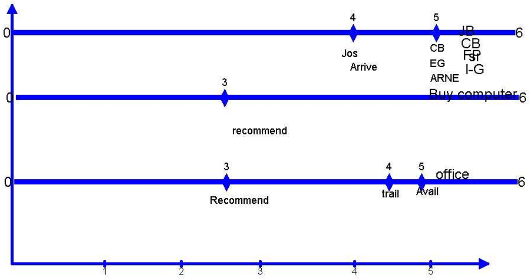


Figure D.8: Phase 1 (ink-only) processing of the g3 Meeting.

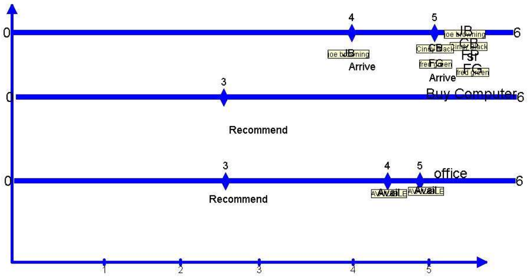


Figure D.9: Phase 2 (ink + speech) processing of the g3 Meeting.

D.7 G4 MEETING

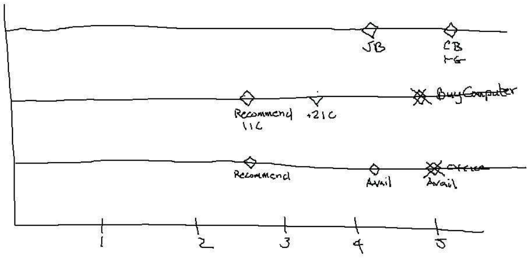


Figure D.10: Ink for the g4 Meeting.

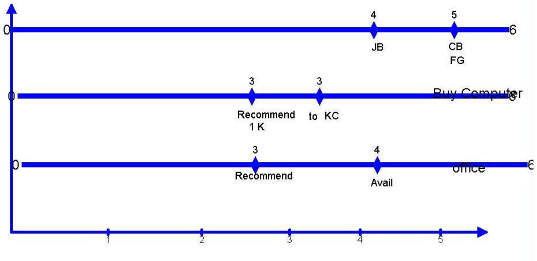


Figure D.11: Phase 1 (ink-only) processing of the g4 Meeting.

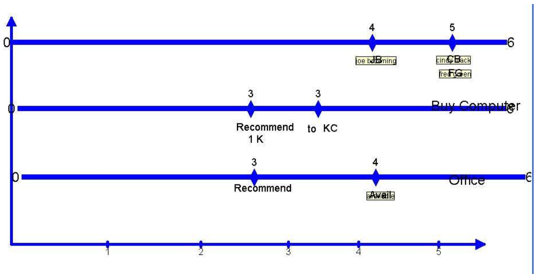


Figure D.12: Phase 2 (ink + speech) processing of the g4 Meeting.

D.8 G5 MEETING

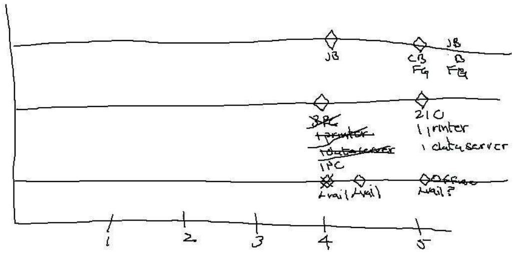


Figure D.13: Ink for the g5 Meeting.

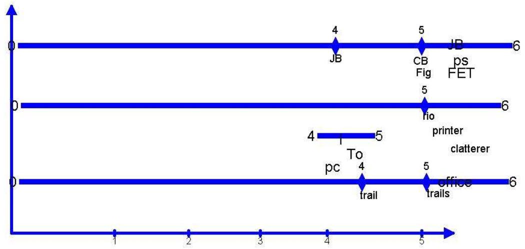


Figure D.14: Phase 1 (ink-only) processing of the g5 Meeting.

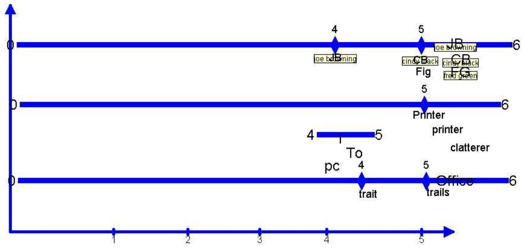


Figure D.15: Phase 2 (ink + speech) processing of the g5 Meeting.

D.8.1 COMBINED SPEECH/GESTURE TRANSCRIPTS

The development set of meetings from the CALO year 2 data set is referred to as the *G* series. The transcript for each of the five meetings is given in the sections below.

The key below explains what is shown on each line of output in the following sections. On *Line-1* there is first the sequence numberings of the utterance or gesture (*9(8)* — note that gestures are within upper and lower equal-lines (e.g. =====), then the start-time and end-time of the utterance in seconds and milliseconds, followed by the file name stem of the utterance, which is used to locate the corresponding lattice file. On *Line-2* is the Speechalyzer recognition transcript, followed on *Line-3* by the hand-annotated transcript, and on *Line-4* the canonical phone sequence for the hand-annotated transcript.

Key Example:

```
Line-1.  9(8):  1116970739.849 1116970741.373 transcript (01050427Z004148):
Line-2.  IS THERE SOME WITH TILL
Line-3.  IS THERE SOME WAY TO TELL
Line-4.  IH Z DH EH R S AH M W EY T IH T EH L
```

D.8.2 G5 FULL TRANSCRIPT

```
Meeting ID: 1117129989000_G5

1(0):  1117130117.731 1117130119.588 transcript (01050926Z113634):


2(1):  1117130120.731 1117130123.131 transcript (01050926Z113639):

YOU KNOW
Y UW N OW

3(2):  1117130123.131 1117130126.484 transcript (01050926Z113647):

I'M GETTING
AY M G EH T IH NG

4(3):  1117130126.484 1117130131.64 transcript (01050926Z113659):
THE SAME RESPONSE WHETHER SOME WAY IT FOR
THE SAME RESPONSE WHETHER IT'S ON MY HEAD OR UH THAT'S
DH IY S EY M R IH S P AA N S W EH DH ER IH T S AA N M AY HH EH D ER DH AE T S

5(4):  1117130132.7 1117130133.84 transcript (01050926Z113717):
THAT'S BETTER
BETTER
B EH T ER

6(5):  1117130133.103 1117130133.884 transcript (01050926Z113719):
GOOD
GOOD
G UH D

7(6):  1117130134.750 1117130136.417 transcript (01050926Z113722):
WELL CASE
OKAY
OW K EY

8(7):  1117130141.131 1117130143.179 transcript (01050926Z113728):
```

```
ALRIGHT ELEMENT
ALRIGHT GENTLEMEN UM
AO L R AY T JH EH N AH L M IH N

9(8):  1117130144.36 1117130145.636 transcript (01050926Z113734):
LET'S GET STARTED
LET'S GET STARTED
L EH T S G EH T S T AA R T AH D

10(9):  1117130145.855 1117130148.26 transcript (01050926Z113738):
AND JUST
AND UH JUST
AE N D JH IH S T

11(10):  1117130148.474 1117130154.103 transcript (01050926Z113743):
QUICKLY ORIENTED RE BUTT IN THERE WILL HAVE FOLKS REPORT OUT HERE WE ARE WITH OR TIMELINE
QUICKLY TO ORIENT EVERYBODY AND THEN WE'LL HAVE FOLKS REPORT OUT HERE WE ARE WITH OUR TIMELINE
K W IH K L IY T UW AO R IY EH N T EH V R IY B AA D IY AE N D DH EH N W IY L HH AE V F OW K S R IH P AO R T AW T HH IH R W IY AA R W IH
TH AW ER T AY M L AY N

== GESTURE ==========================================================
1:  1117130152.222 1117130155.738
[xy_axis(prob(0, 0.957)), line(prob(6, 0.748)), writing(prob(13, 0.55), text(1)), writing(prob(21, 0.378), text(L)), writing(prob(25, 0.318),
text(z)), writing(prob(25, 0.312), text(k)), writing(prob(26, 0.306), text(l)), writing(prob(26, 0.3), text(t)), writing(prob(26, 0.294),
text(i)), writing(prob(27, 0.288), text(r)), arrow(prob(30, 0.248), vh(305, -6), vh(797, 901), 334)]
=====================================================================

== GESTURE ==========================================================
2:  1117130156.972 1117130157.191
[tick(prob(8, 0.672)), writing(prob(13, 0.53), text(1)), writing(prob(24, 0.33), text(I)), writing(prob(24, 0.324), text(l)), line(prob(26,
0.303)), writing(prob(26, 0.3), text(i)), writing(prob(27, 0.282), text(K)), writing(prob(28, 0.276), text(M)), arrow(prob(47, 0.116),
vh(655, 403), vh(702, 393), 258)]
=====================================================================

12(11):  1117130157.45 1117130158.636 transcript (01050926Z113802):
WEEK WHEN
WEEK ONE
W IY K HH W AH N

== GESTURE ==========================================================
3:  1117130157.566 1117130157.753
[writing(prob(13, 0.53), text(1)), tick(prob(14, 0.512)), writing(prob(24, 0.33), text(I)), writing(prob(25, 0.312), text(l)), writing(prob(26,
0.3), text(i)), writing(prob(27, 0.282), text(L)), writing(prob(28, 0.276), text(K)), line(prob(38, 0.171))]
=====================================================================

== GESTURE ==========================================================
4:  1117130158.363 1117130158.769
[tick(prob(6, 0.729)), writing(prob(13, 0.53), text(1)), writing(prob(24, 0.33), text(I)), writing(prob(25, 0.312), text(l)), writing(prob(26,
0.3), text(i)), writing(prob(27, 0.288), text(y)), writing(prob(27, 0.282), text(K)), line(prob(28, 0.28)), writing(prob(28, 0.276), text(M))]
=====================================================================

== GESTURE ==========================================================
5:  1117130159.35 1117130159.535
[writing(prob(13, 0.55), text(2)), writing(prob(14, 0.52), text(1)), writing(prob(15, 0.5), text(3)), tick(prob(22, 0.365)), writing(prob(24,
0.324), text(z)), writing(prob(25, 0.318), text(Z)), writing(prob(26, 0.306), text(L)), writing(prob(26, 0.294), text(i)), writing(prob(27,
0.288), text(a)), writing(prob(27, 0.282), text(s)), writing(prob(28, 0.276), text(B)), line(prob(41, 0.154))]
=====================================================================

13(12):  1117130159.274 1117130160.360 transcript (01050926Z113806):
TO
WEEK TWO
W IY K T UW

== GESTURE ==========================================================
6:  1117130160.316 1117130160.863
```

```
[tick(prob(10, 0.627)), writing(prob(13, 0.53), text(1)), line(prob(24, 0.333)), writing(prob(24, 0.33), text(I)), writing(prob(25, 0.312),
text(l)), writing(prob(26, 0.3), text(i)), writing(prob(27, 0.282), text(V)), writing(prob(28, 0.276), text(y)), arrow(prob(48, 0.111),
vh(657, 676), vh(711, 678), 274)]
=======================================================================

== GESTURE ============================================================
7:  1117130161.160 1117130161.753
[writing(prob(10, 0.63), text(3)), writing(prob(25, 0.318), text(z)), writing(prob(25, 0.312), text(g)), writing(prob(26, 0.306), text(J)),
writing(prob(26, 0.3), text(s)), writing(prob(26, 0.294), text(I)), tick(prob(28, 0.272)), line(prob(45, 0.125))]
=======================================================================

14(13):  1117130161.179 1117130162.636 transcript (01050926Z113809):
THE FOR THE
WEEK THREE
W IY K TH R IY

== GESTURE ============================================================
8:  1117130162.457 1117130162.941
[tick(prob(8, 0.683)), writing(prob(13, 0.53), text(1)), writing(prob(24, 0.33), text(I)), writing(prob(25, 0.312), text(l)), writing(prob(26,
0.3), text(i)), writing(prob(27, 0.282), text(y)), line(prob(28, 0.278)), writing(prob(28, 0.276), text(K))]
=======================================================================

== GESTURE ============================================================
9:  1117130163.253 1117130164.753
[writing(prob(10, 0.63), text(4)), writing(prob(24, 0.33), text(a)), writing(prob(24, 0.324), text(q)), arrow(prob(42, 0.146), vh(710,
804), vh(747, 814), 288), line(prob(44, 0.131))]
=======================================================================

15(14):  1117130163.407 1117130167.112 transcript (01050926Z113813):
WHICH SO ONE GO FOR THE SYSTEM THAT BUT THE TIMELINE FOR



== GESTURE ============================================================
10:  1117130165.738 1117130166.425
[tick(prob(8, 0.677)), writing(prob(14, 0.52), text(1)), writing(prob(17, 0.46), text(No)), writing(prob(24, 0.33), text(I)), writing(prob(25,
0.318), text(l)), writing(prob(26, 0.3), text(i)), line(prob(27, 0.292)), writing(prob(27, 0.288), text(K)), writing(prob(27, 0.282), text(N)),
arrow(prob(46, 0.12), vh(675, 983), vh(720, 992), 284)]
=======================================================================

== GESTURE ============================================================
11:  1117130166.847 1117130167.910
[writing(prob(10, 0.63), text(5)), writing(prob(13, 0.54), text(s-)), writing(prob(13, 0.53), text(g-)), writing(prob(24, 0.33), text(T)),
writing(prob(25, 0.312), text(s))]
=======================================================================

16(15):  1117130167.712 1117130169.7 transcript (01050926Z113828):
INTO TO YOU
HAND IT TO YOU
HH AE N D IH T T IH Y UW

17(16):  1117130169.350 1117130170.550 transcript (01050926Z113831):
ALRIGHT
ALRIGHT
AO L R AY T

== GESTURE ============================================================
12:  1117130169.879 1117130171.957
[line(prob(2, 0.883)), writing(prob(12, 0.56), text(sr)), writing(prob(13, 0.55), text(Mr)), writing(prob(13, 0.54), text(nr)), writing(prob(13,
0.53), text(mr)), writing(prob(14, 0.51), text(ir)), writing(prob(15, 0.5), text(er)), writing(prob(15, 0.49), text(ire)), writing(prob(16,
0.48), text(ira)), arrow(prob(21, 0.381), vh(187, 163), vh(204, 1111), 360), xy_axis(prob(29, 0.267)), writing(prob(36, 0.187), text())]
=======================================================================

18(17):  1117130172.17 1117130173.293 transcript (01050926Z113834):
WE HAVE
WE HAVE
```

```
W IY HH AE V

== GESTURE =============================================================
13:  1117130173.644 1117130174.972
[writing(prob(13, 0.55), text(IB)), writing(prob(13, 0.53), text(BB)), writing(prob(14, 0.52), text(SB)), writing(prob(14, 0.51), text(3B)),
writing(prob(15, 0.5), text(Jos)), writing(prob(15, 0.49), text(JPs)), writing(prob(16, 0.48), text(Ips)), writing(prob(16, 0.47), text(sis)),
writing(prob(17, 0.46), text(jpg)), writing(prob(24, 0.324), text(B)), diamond(prob(46, 0.12))]
=======================================================================

19(18):  1117130174.17 1117130175.341 transcript (01050926Z113838):
JOB RUNNING
JOE_BROWNING
JH OW B R AW N IH NG

== GESTURE =============================================================
14:  1117130176.160 1117130177.738
[writing(prob(25, 0.312), text(a B)), writing(prob(27, 0.288), text(i B)), writing(prob(28, 0.276), text(r B)), diamond(prob(29, 0.259)),
arrow(prob(44, 0.131), vh(222, 1049), vh(247, 1075), 320)]
=======================================================================

20(19):  1117130176.207 1117130180.760 transcript (01050926Z113841):
THIS IN THE BLACK AND FRED'S GREEN
UH CINDY_BLACK AND FRED_GREEN
S IH N D IY B L AE K AE N D F R EH D G R IY N

== GESTURE =============================================================
15:  1117130178.285 1117130180.426
[writing(prob(13, 0.55), text(Fey)), writing(prob(13, 0.54), text(ECT)), writing(prob(13, 0.53), text(Fee)), writing(prob(14, 0.52), text(Foy)),
writing(prob(14, 0.51), text(Frey)), writing(prob(15, 0.5), text(t-e, )), writing(prob(15, 0.49), text(t-er)), writing(prob(16, 0.48),
text(t-el)), writing(prob(16, 0.47), text(r-er)), writing(prob(17, 0.46), text(t-or))]
=======================================================================

21(20):  1117130181.55 1117130184.84 transcript (01050926Z113854):
AND JOB RUNNING IS ARRIVING HERE STILL
AND JOE_BROWNING IS ARRIVING HERE STILL
AE N D JH OW B R AW N IH NG IH Z ER AY V IH NG HH IH R S T IH L

== GESTURE =============================================================
16:  1117130182.269 1117130184.457
[diamond(prob(5, 0.79)), writing(prob(14, 0.51), text(9)), writing(prob(15, 0.5), text(4)), writing(prob(16, 0.48), text(47)), writing(prob(16,
0.47), text(67)), writing(prob(24, 0.324), text(H)), writing(prob(26, 0.294), text(A)), arrow(prob(38, 0.172), vh(161, 824), vh(214, 820),
266)]
=======================================================================

22(21):  1117130184.207 1117130186.7 transcript (01050926Z113903):
BUT STORED WEEK FOR
AT THE START OF WEEK FOUR
AE T DH IY S T AA R T AH V W IY K F AO R

23(22):  1117130186.369 1117130188.36 transcript (01050926Z113908):
AND
AND UM
AE N D

== GESTURE =============================================================
17:  1117130187.801 1117130189.113
[writing(prob(9, 0.65), text(JB)), writing(prob(9, 0.64), text(IB)), writing(prob(10, 0.63), text(SB)), writing(prob(11, 0.6), text(3B)),
writing(prob(14, 0.52), text(TB)), writing(prob(14, 0.51), text(FB)), writing(prob(15, 0.49), text(JOB)), writing(prob(16, 0.48), text(JIB)),
writing(prob(17, 0.46), text(JAB)), diamond(prob(25, 0.317))]
=======================================================================

24(23):  1117130190.960 1117130192.464 transcript (01050926Z113912):
CINDY'S GREEN
CINDY_GREEN
S IH N D IY G R IY N
```

```
== GESTURE ============================================================
18:  1117130191.51 1117130192.988
[diamond(prob(6, 0.751)), writing(prob(13, 0.55), text(4)), writing(prob(13, 0.54), text(0)), writing(prob(16, 0.48), text(Ro)), writing(prob(16,
0.47), text(Ay)), writing(prob(17, 0.46), text(AD)), writing(prob(25, 0.318), text(H)), writing(prob(25, 0.312), text(x)), writing(prob(26,
0.306), text(D)), writing(prob(26, 0.3), text(A)), writing(prob(26, 0.294), text(R)), arrow(prob(42, 0.146), vh(175, 984), vh(215, 991),
281)]
=======================================================================

25(24):  1117130192.464 1117130194.293 transcript (01050926Z113916):
AND FRED'S BLACK
AND FRED_BLACK
AE N D F R EH D B L AE K

== GESTURE ============================================================
19:  1117130193.473 1117130195.941
[writing(prob(7, 0.7), text(CB)), writing(prob(11, 0.59), text(cb)), writing(prob(13, 0.55), text(as)), writing(prob(13, 0.54), text(cos)),
writing(prob(13, 0.53), text(eb)), writing(prob(14, 0.52), text(lb)), writing(prob(14, 0.51), text(cis)), writing(prob(15, 0.5), text(eB)),
writing(prob(16, 0.48), text(ib)), diamond(prob(26, 0.3))]
=======================================================================

26(25):  1117130194.312 1117130197.417 transcript (01050926Z113922):
PARSERS SUNDAY BLACK INFER AGREEING
I'M SORRY CINDY_BLACK AND FRED_GREEN
AY M S AA R IY S IH N D IY B L AE K AE N D F R EH D G R IY N

== GESTURE ============================================================
20:  1117130196.348 1117130198.113
[writing(prob(9, 0.64), text(Foy)), writing(prob(13, 0.55), text(Fy)), writing(prob(13, 0.53), text(Fop)), writing(prob(14, 0.52), text(Fs)),
writing(prob(14, 0.51), text(Foe)), writing(prob(15, 0.5), text(Ftp)), writing(prob(15, 0.49), text(fa)), writing(prob(16, 0.48), text(Ft.)),
writing(prob(16, 0.47), text(t-g)), writing(prob(17, 0.46), text(t-a)), diamond(prob(46, 0.122))]
=======================================================================

27(26):  1117130197.512 1117130200.684 transcript (01050926Z113931):
OR ARRIVING HERE TO STORE THAT WE FIVE
ARE ARRIVING HERE AT THE START OF WEEK FIVE
AA R ER AY V IH NG HH IH R AE T DH IY S T AA R T AH V W IY K F AY V

28(27):  1117130200.684 1117130204.560 transcript (01050926Z113943):
WHERE RIGHT HERE AT AT THE START OF WEEK FOURIER
WE'RE RIGHT HERE AT THE START OF WEEK THREE
W IY R R AY T HH IH R AE T DH AH S T AA R T AH V W IY K TH R IY

== GESTURE ============================================================
21:  1117130203.738 1117130207.4
[line(prob(2, 0.892)), arrow(prob(19, 0.419), vh(372, 168), vh(344, 1088), 1), writing(prob(26, 0.306), text(e)), writing(prob(26, 0.3),
text(J)), writing(prob(26, 0.294), text(T)), writing(prob(27, 0.288), text(G)), writing(prob(27, 0.282), text(L)), writing(prob(28, 0.276),
text(I)), xy_axis(prob(29, 0.262))]
=======================================================================

29(28):  1117130207.731 1117130209.426 transcript (01050926Z113955):
AND
AND UM
AE N D

== GESTURE ============================================================
22:  1117130212.426 1117130215.770
[diamond(prob(5, 0.773)), writing(prob(13, 0.53), text(97)), writing(prob(14, 0.52), text(ay)), writing(prob(24, 0.324), text(D)), arrow(prob(40,
0.158), vh(330, 801), vh(375, 798), 267)]
=======================================================================

30(29):  1117130212.674 1117130213.931 transcript (01050926Z113959):
SO FAR
SO FAR
S OW F AA R

31(30):  1117130214.684 1117130220.331 transcript (01050926Z114002):
```

```
CHECKERS TOLD US THAT BUYING THE START A WEEK FOR WE WILL HAVE THREE COMPUTERS
JACK HAS TOLD US THAT BY THE START OF WEEK FOUR WE WILL HAVE THREE COMPUTERS
JH AE K HH AH Z T OW L D AH S DH AE T B AY DH IY S T AA R T AH V W IY K F AO R W IY W IH L HH AE V TH R IY K AH M P Y UW T ER Z

== GESTURE ===========================================================
23:  1117130218.145 1117130221.645
[writing(prob(13, 0.55), text(3pc)), writing(prob(13, 0.54), text(3K)), writing(prob(13, 0.53), text(37K)), writing(prob(14, 0.52), text(32K)),
writing(prob(14, 0.51), text(34K)), writing(prob(15, 0.5), text(377K)), writing(prob(15, 0.49), text(Jpe)), writing(prob(16, 0.48), text(Jpg)),
writing(prob(16, 0.47), text(spe)), writing(prob(17, 0.46), text(Jpi))]
======================================================================

32(31):  1117130222.179 1117130224.588 transcript (01050926Z114017):
AND ONE PRINTER
AND ONE PRINTER
AE N D W AH N P R IH N ER

== GESTURE ===========================================================
24:  1117130223.629 1117130228.20
[writing(prob(13, 0.55), text(sprinter)), writing(prob(13, 0.54), text(printer)), writing(prob(13, 0.53), text(Printer)), writing(prob(14,
0.52), text(printers)), writing(prob(15, 0.5), text(Imprinter)), writing(prob(15, 0.49), text(Printers)), writing(prob(16, 0.47), text(sprint-or))]
======================================================================

== GESTURE ===========================================================
25:  1117130228.676 1117130237.395
[writing(prob(13, 0.55), text(1datascrver)), writing(prob(13, 0.53), text(1dataserver)), writing(prob(14, 0.52), text(data server)), writing(prob(14,
0.51), text(1datasorver)), writing(prob(15, 0.5), text(1data server)), writing(prob(15, 0.49), text(1datasaver)), writing(prob(16, 0.48),
text(rata server)), writing(prob(16, 0.47), text(caterer)), writing(prob(17, 0.46), text(Hakata server))]
======================================================================

33(32):  1117130228.798 1117130230.188 transcript (01050926Z114023):
IN ONE
AND ONE
AE N D W AH N

34(33):  1117130230.331 1117130232.645 transcript (01050926Z114027):
DATA SERVER
UH DATA SERVER
D EY T AH S ER V ER

35(34):  1117130250.579 1117130252.998 transcript (01050926Z114032):
OKAY SO OLD
OKAY SO
OW K EY S OW

36(35):  1117130254.64 1117130255.826 transcript (01050926Z114040):
LET'S LISTEN TO BUILD
LET'S LISTEN TO BILL
L EH T S L IH S AH N T AH B IH L

37(36):  1117130265.541 1117130266.379 transcript (01050926Z114045):




38(37):  1117130289.588 1117130292.436 transcript (01050926Z114048):
SO BY BY THE START OF WEEK FIVE
SO BY BY THE START OF WEEK FIVE
S OW B AY B AY DH AH S T AA R T AH V W IY K F AY V

39(38):  1117130295.341 1117130297.150 transcript (01050926Z114056):
SO
SO
S OW

40(39):  1117130297.579 1117130301.579 transcript (01050926Z114103):
CHECK IT I UNDERSTAND YOU TO SAY THAT THAT THE
```

```
JACK DID I UNDERSTAND YOU TO SAY THAT THE UM
JH AE K D IH D AY AH N D ER S T AE N D Y UW T IH S EY DH AH T DH IY

41(40):  1117130301.912 1117130305.798 transcript (01050926Z114115):
THAT THERE WAS A DELAY IN THE TWO P_C'S WHICH WE HAD TO ORDER
THAT THERE WAS A DELAY IN THE TWO P_C'S WHICH WE HAD TO ORDER
DH AH T DH EH R W AH Z AH D IH L EY IH N DH IY T UW P IY S IY Z W IH CH W IY HH AE D T UW AO R D ER

42(41):  1117130335.512 1117130336.703 transcript (01050926Z114128):
ALRIGHT
ALRIGHT
AO L R AY T

43(42):  1117130336.750 1117130339.122 transcript (01050926Z114131):
SO LET'S DO THIS
SO LET'S DO THIS
S OW L EH T S D UW DH IH S

== GESTURE ============================================================
26:  1117130336.788 1117130337.803
[cross(prob(13, 0.538)), writing(prob(14, 0.52), text(Th)), writing(prob(14, 0.51), text(th)), writing(prob(15, 0.5), text(Xx)), writing(prob(15,
0.49), text(xx)), writing(prob(16, 0.47), text(Ta)), writing(prob(21, 0.378), text(x)), writing(prob(24, 0.33), text(X)), writing(prob(25,
0.318), text(y)), writing(prob(27, 0.288), text(t)), arrow(prob(29, 0.256), vh(413, 768), vh(397, 839), 11), diamond(prob(34, 0.211)),
line(prob(38, 0.175))]
=======================================================================

== GESTURE ============================================================
27:  1117130338.256 1117130338.694
[line(prob(13, 0.531)), arrow(prob(30, 0.25), vh(442, 785), vh(438, 897), 1), tick(prob(44, 0.132))]
=======================================================================

== GESTURE ============================================================
28:  1117130338.944 1117130339.319
[line(prob(13, 0.539)), writing(prob(25, 0.312), text(I)), writing(prob(26, 0.294), text(S)), writing(prob(27, 0.288), text(i)), writing(prob(27,
0.282), text(s)), writing(prob(28, 0.276), text(T)), arrow(prob(30, 0.25), vh(418, 888), vh(478, 777), 206), xy_axis(prob(35, 0.203)), tick(prob(44,
0.134))]
=======================================================================

== GESTURE ============================================================
29:  1117130339.663 1117130340.881
[writing(prob(12, 0.56), text(To)), writing(prob(25, 0.318), text(i)), writing(prob(25, 0.312), text(x)), arrow(prob(26, 0.303), vh(495,
788), vh(481, 932), 5), writing(prob(26, 0.3), text(T)), writing(prob(26, 0.294), text(F)), writing(prob(27, 0.288), text(t)), line(prob(34,
0.208)), xy_axis(prob(49, 0.105))]
=======================================================================

44(43):  1117130340.674 1117130343.503 transcript (01050926Z114138):
AND WE HAVE ONE P_C
AND WE HAVE ONE P_C
AE N D W IY HH AE V W AH N P IY S IY

== GESTURE ============================================================
30:  1117130341.772 1117130343.678
[writing(prob(9, 0.65), text(pc)), writing(prob(10, 0.63), text(pl)), writing(prob(13, 0.54), text(1pc)), writing(prob(14, 0.52), text(po)),
writing(prob(14, 0.51), text(1PC)), writing(prob(15, 0.5), text(1Pc)), writing(prob(15, 0.49), text(pl.)), writing(prob(16, 0.47), text(pee)),
diamond(prob(35, 0.2))]
=======================================================================

45(44):  1117130344.369 1117130346.750 transcript (01050926Z114146):
THAT WEEK FOR FOR A JOB RUNNING
AT WEEK FOUR FOR JOE_BROWNING
AE T W IY K F AO R F R ER JH OW B R AW N IH NG

46(45):  1117130346.950 1117130349.988 transcript (01050926Z114153):
AND THEN IT TO START A WEEK BIAS
AND THEN AT THE START OF WEEK FIVE
```

```
AE N D DH EH N AE T DH AH S T AA R T AH V W IY K F AY V

== GESTURE ============================================================
31:  1117130348.210 1117130350.616
[diamond(prob(9, 0.649)), writing(prob(12, 0.56), text(Ay)), writing(prob(13, 0.55), text(4)), writing(prob(14, 0.51), text(9)), writing(prob(16,
0.48), text(Ra)), writing(prob(16, 0.47), text(Ah)), writing(prob(24, 0.324), text(H)), writing(prob(25, 0.312), text(A)), writing(prob(26,
0.3), text(R)), writing(prob(26, 0.294), text(D)), arrow(prob(40, 0.157), vh(322, 992), vh(370, 994), 273)]
=======================================================================

47(46):  1117130349.988 1117130351.198 transcript (01050926Z114203):
WE WILL HAVE
WE WILL HAVE
W IY W AH L HH AE V

== GESTURE ============================================================
32:  1117130351.38 1117130353.69
[writing(prob(10, 0.63), text(210)), writing(prob(13, 0.54), text(xo)), writing(prob(13, 0.53), text(40)), writing(prob(14, 0.52), text(210)),
writing(prob(14, 0.51), text(HO)), writing(prob(15, 0.5), text(2lo)), writing(prob(15, 0.49), text(210)), writing(prob(16, 0.48), text(218)),
writing(prob(16, 0.47), text(rio))]
=======================================================================

48(47):  1117130351.484 1117130353.55 transcript (01050926Z114207):
TWO P_C'S
THE TWO P_C'S
DH AH T UW P IY S IY Z

49(48):  1117130353.303 1117130356.17 transcript (01050926Z114210):
FOR SUNDAY BLACK AND FRED'S GREEN
FOR CINDY_BLACK AND FRED_GREEN
F R ER S IH N D IY B L AE K AE N D F R EH D G R IY N

50(49):  1117130356.17 1117130357.674 transcript (01050926Z114217):
ONE ONE PRINTER
WE'LL HAVE ONE PRINTER
W IY L HH AE V W AH N P R IH N ER

== GESTURE ============================================================
33:  1117130356.382 1117130356.663
[writing(prob(13, 0.54), text(1)), writing(prob(24, 0.33), text(I)), writing(prob(25, 0.318), text(l)), writing(prob(26, 0.3), text(i)),
writing(prob(27, 0.288), text(r)), writing(prob(27, 0.282), text(V)), writing(prob(28, 0.276), text(y)), tick(prob(34, 0.205)), line(prob(36,
0.191))]
=======================================================================

== GESTURE ============================================================
34:  1117130357.85 1117130361.257
[writing(prob(9, 0.65), text(printer)), writing(prob(11, 0.59), text(printers)), writing(prob(13, 0.54), text(grunter)), writing(prob(13,
0.53), text(painter)), writing(prob(14, 0.52), text(granter)), writing(prob(14, 0.51), text(punter)), writing(prob(15, 0.5), text(irinter)),
writing(prob(16, 0.48), text(lrinter)), writing(prob(16, 0.47), text(trent-er)), writing(prob(17, 0.46), text(trial-er))]
=======================================================================

51(50):  1117130361.455 1117130364.703 transcript (01050926Z114222):
AND WHEN DO IS SERVER SIDE RIGHT
AND ONE DATA SERVER IS THAT RIGHT
AE N D W AH N D EY T AH S ER V ER IH Z DH AE T R AY T

== GESTURE ============================================================
35:  1117130362.69 1117130369.147
[writing(prob(10, 0.63), text(clatterer)), writing(prob(11, 0.59), text(Clatterer)), writing(prob(13, 0.55), text(Idatuserver)), writing(prob(14,
0.51), text(iclatuserver)), writing(prob(16, 0.47), text(idatuserver)), writing(prob(17, 0.46), text(idolater server))]
=======================================================================

52(51):  1117130369.560 1117130370.617 transcript (01050926Z114232):
ALRIGHT
ALRIGHT
AO L R AY T
```

```
53(52):  1117130371.445 1117130374.531 transcript (01050926Z114234):
SO IT'S GETTING TIRED BUT IT'S STILL DOABLE
SO IT'S GETTING TIGHTER BUT IT'S STILL DOABLE
S OW IH T S G IH T IH NG T AY T ER B AH T IH T S S T IH L D UW AH B AH L

54(53):  1117130376.245 1117130380.226 transcript (01050926Z114243):
ALRIGHT SO LET'S LOOK AT OUR OFFICE TIMELINE
ALRIGHT SO LET'S LOOK AT OUR OFFICE TIMELINE
AO L R AY T S OW L EH T S L UH K AE T AW ER AO F AH S T AY M L AY N

== GESTURE ============================================================
36:  1117130376.835 1117130379.585
[line(prob(3, 0.86)), writing(prob(9, 0.65), text(mr)), writing(prob(13, 0.54), text(win)), writing(prob(13, 0.53), text(men)), writing(prob(14,
0.52), text(nn)), writing(prob(14, 0.51), text(ms)), writing(prob(15, 0.5), text(won)), writing(prob(15, 0.49), text(wen)), writing(prob(16,
0.48), text(min)), writing(prob(16, 0.47), text(wir)), writing(prob(17, 0.46), text(rm)), arrow(prob(19, 0.413), vh(564, 158), vh(564,
1096), 0), xy_axis(prob(45, 0.125))]
=======================================================================

== GESTURE ============================================================
37:  1117130380.413 1117130384.163
[writing(prob(7, 0.7), text(office)), writing(prob(9, 0.65), text(Office)), writing(prob(12, 0.56), text(off ice)), writing(prob(13, 0.54),
text(offrce)), writing(prob(13, 0.53), text(off-ce)), writing(prob(14, 0.52), text(offie)), writing(prob(14, 0.51), text(Offie)), writing(prob(15,
0.5), text(offin)), writing(prob(15, 0.49), text(of-fia)), writing(prob(16, 0.48), text(off Ice))]
=======================================================================

55(54):  1117130381.722 1117130384.312 transcript (01050926Z114256):
BY A DON'T BELIEVE SO
I DON'T BELIEVE SO
AY D OW N B IH L IY V S OW

56(55):  1117130385.874 1117130387.74 transcript (01050926Z114304):
THANKS VERY MUCH
THANKS VERY MUCH
TH AE NG K S V EH R IY M AH CH

57(56):  1117130387.360 1117130391.817 transcript (01050926Z114308):
OKAY SO CORD OR OFFICE TIMELINE WE'RE GONNA HEAVILY OFFICE SPACE
OKAY SO ACCORDING TO OUR OFFICE TIMELINE WE WERE GOING TO HAVE ALL THE OFFICE SPACE
OW K EY S OW AH K AO R D IH NG T AH AW ER AO F AH S T AY M L AY N W IY W ER G OW IH N T IH HH AE V AO L DH IY AO F AH S S P EY S

== GESTURE ============================================================
38:  1117130391.382 1117130393.507
[diamond(prob(5, 0.791)), writing(prob(11, 0.6), text(is)), writing(prob(12, 0.57), text(as)), writing(prob(13, 0.53), text(4)), writing(prob(14,
0.51), text(0)), writing(prob(24, 0.33), text(A)), writing(prob(24, 0.324), text(D)), writing(prob(26, 0.294), text(d)), arrow(prob(43,
0.137), vh(550, 813), vh(584, 810), 263)]
=======================================================================

58(57):  1117130391.979 1117130394.303 transcript (01050926Z114321):
TAKING CARE OF RIGHT HERE
TAKEN CARE OF RIGHT HERE
T EY K AH N K EH R AH V R AY T HH IH R

== GESTURE ============================================================
39:  1117130395.710 1117130398.804
[writing(prob(10, 0.62), text(Avail)), writing(prob(13, 0.55), text(trail)), writing(prob(13, 0.54), text(Frail)), writing(prob(13, 0.53),
text(frail)), writing(prob(14, 0.51), text(t-rail)), writing(prob(15, 0.5), text(f-rail)), writing(prob(15, 0.49), text(Ho-rail)), writing(prob(16,
0.48), text(to-rail)), writing(prob(16, 0.47), text(f-Vail)), writing(prob(17, 0.46), text(Ah-rail))]
=======================================================================

59(58):  1117130399.322 1117130400.493 transcript (01050926Z114328):
AND
AND
AE N D

60(59):  1117130401.169 1117130403.93 transcript (01050926Z114332):
JOHN:Person_Name TIMIT OR STAY WITH THIS
```

```
JOHN TELL ME WHERE WE STAND WITH THIS
JH AA N T EH L M IY W EH R W IY S T AE N D W IH DH DH IH S

== GESTURE ============================================================
40:  1117130410.492 1117130411.867
[writing(prob(11, 0.59), text(xo)), writing(prob(12, 0.57), text(XO)), writing(prob(13, 0.54), text(X.)), writing(prob(13, 0.53), text(Xo)),
writing(prob(14, 0.52), text(Xl)), writing(prob(15, 0.5), text(x.)), writing(prob(16, 0.48), text(xl)), writing(prob(21, 0.378), text(x)),
writing(prob(24, 0.33), text(X)), diamond(prob(25, 0.308)), writing(prob(26, 0.306), text(K)), arrow(prob(39, 0.166), vh(544, 815), vh(588,
806), 257), line(prob(46, 0.119))]
=======================================================================

== GESTURE ============================================================
41:  1117130413.304 1117130415.851
[diamond(prob(4, 0.811)), writing(prob(13, 0.55), text(4)), writing(prob(13, 0.54), text(0)), writing(prob(25, 0.318), text(x)), writing(prob(25,
0.312), text(J)), writing(prob(26, 0.306), text(n)), writing(prob(26, 0.3), text(D)), writing(prob(26, 0.294), text(A)), writing(prob(27,
0.288), text(R)), arrow(prob(45, 0.124), vh(548, 868), vh(580, 878), 291)]
=======================================================================

== GESTURE ============================================================
42:  1117130416.820 1117130420.945
[writing(prob(9, 0.65), text(trail)), writing(prob(9, 0.64), text(Avail)), writing(prob(10, 0.62), text(frail)), writing(prob(11, 0.6),
text(Frail)), writing(prob(11, 0.58), text(t-rail)), writing(prob(13, 0.53), text(trait)), writing(prob(14, 0.51), text(trill)), writing(prob(17,
0.46), text(l-viii))]
=======================================================================

61(60):  1117130422.455 1117130423.369 transcript (01050926Z114339):
RIGHT
RIGHT
R AY T

62(61):  1117130502.845 1117130505.503 transcript (01050926Z114343):
THING I I A IF WE'RE GOING TO PUT
THE THING I I IF WE'RE GOING TO PUT
DH IY TH IH NG AY AY IH F W ER G OW IH N T IH P UH T

63(62):  1117130505.845 1117130508.484 transcript (01050926Z114351):
WRIST BOOT THEM UP AND TO OFFICES
WE'RE GOING TO SPLIT THEM UP INTO TWO OFFICES
W IH R G OW IH N T IH S P L IH T DH EH M AH P IH N T AH T UW AO F AH S AH Z

64(63):  1117130509.131 1117130516.7 transcript (01050926Z114358):
JACKED NATIVE VARY COMPILING KEYS LAST MEETING THAT WE DON'T WANT TO PUT THEM THE SAME OFFICE WITH
UM JACK MADE A VERY COMPELLING CASE LAST MEETING THAT WE DON'T WANT TO PUT THEM IN THE SAME OFFICE WITH
JH AE K M EY D AH V EH R IY K AH M P EH L IH NG K EY S L AE S M IY T IH NG DH AH T W IY D OW N W AA N T T IH P UH T DH AH M IH N DH AH
S EY M AO F AH S W IH DH

65(64):  1117130516.7 1117130518.988 transcript (01050926Z114419):
THE READ SERVER'S PARTICULAR AND THE PRINTER
THE RAID SERVER IN PARTICULAR AND THE PRINTER
DH IY R EY D S ER V ER IH N P ER T IH K Y AH L ER AE N D DH AH P R IH N T ER

66(65):  1117130519.322 1117130522.931 transcript (01050926Z114427):
SO ALWAYS NOW WE HAVE ARE REQUIREMENT FOR FREE SPACES
SO WE NOW WE HAVE A REQUIREMENT FOR THREE SPACES
S OW W IY N AW W IY HH AE V AH R IH K W AY R M AH N T F ER TH R IY S P EY S AH Z

67(66):  1117130523.245 1117130527.960 transcript (01050926Z114437):
WHAT A SINGLE OFFICE THE DOUBLE OFFICE AND EQUIPMENT FOR
WE WANT A SINGLE OFFICE A DOUBLE OFFICE AND AN EQUIPMENT ROOM
W IY W AA N T AH S IH NG G AH L AO F AH S AH D AH B AH L AO F AH S AE N D AE N IH K W IH P M AH N T R UW M

68(67):  1117130528.141 1117130533.760 transcript (01050926Z114452):
AND I STILL WANT TO HAVE THOSE PEOPLE IN IN ROLL IT TO PROXIMITY TOMORROW OFFICE
AND I'D STILL WANT TO HAVE THOSE PEOPLE IN IN RELATIVE PROXIMITY TO MY OFFICE
AE N D AY D S T IH L W AA N T T IH HH AE V DH OW Z P IY P AH L IH N IH N R EH L AH T IH V P R AA K S IH M AH T IY T AH M AY AO F AH S
```

```
69(68):  1117130535.293 1117130547.264 transcript (01050926Z114510):
ANY OTHER THING THAT WE NEED TO CONSIDER IS HOW WE'RE GOING TO SPLIT THEM UP BECAUSE THE THE TELEPHONE DROPS AND WE PUT THE DROPS IN THE
ONE MAKE SURE THE NUMBERS ASSIGNED STAY WITH THE PERSON'S NAME
UH AND THE OTHER THING WE NEED TO CONSIDER IS HOW WE'RE GOING TO SPLIT THEM UP BECAUSE OF THE TELEPHONE DROPS WHEN WE PUT THE DROPS IN
WE WANNA MAKE SURE THE NUMBERS ASSIGNED STAY WITH THE PERSON'S NAME
AH N D DH IY AH DH ER TH IH NG W IY N IY D T AH K AH N S IH D ER IH Z HH AW W ER G OW IH N T AH S P L IH T DH EH M AH P B IH K AH Z AH
V DH AH T EH L AH F OW N D R AA P S W EH N W IY P UH T DH IY D R AA P S IH N W IY W AA N AH M EY K SH UH R DH IY N AH M B ER Z AH S AY
N D S T EY W IH TH DH AH P ER S AH N Z N EY M

70(69):  1117130547.341 1117130556.64 transcript (01050926Z114550):
SO WE DON'T GO THROUGH BUDGET DIRECTORY CHANGES RANDOM PEOPLE TELEPHONES AND SO WHAT THAT'S YOUR PROBLEM I I'M SURE YOU GUYS CAN WORK THAT
THAT'LL
SO WE DON'T GO THROUGH A BUNCH OF DIRECTORY CHANGES TRYING TO MOVE PEOPLE AND TELEPHONES AND SO ON THAT'S YOUR PROBLEM I I'M SURE YOU GUYS
CAN WORK THAT OUT
S OW W IY D OW N G OW TH R UW AH B AH N CH AH V D ER EH K T ER IY CH EY N JH IH Z T R AY IH NG T UW M UW V P IY P AH L AH N D T EH L AH
F OW N Z AH N D S OW AA N DH AE T S Y UH R P R AA B L AH M AY AY M SH UH R Y UW G AY Z K AH N W ER K DH AE T AW T

71(70):  1117130556.331 1117130563.36 transcript (01050926Z114622):
SO WHEN DO YOU THINK WILL KNOW OLD ABOUT THE THE THE SECOND OFFICE WILL BE STUCK IN KNOWS EVERYBODY
SO WHEN DO YOU THINK WE'LL KNOW ABOUT THE UH THE SECOND OFFICE OR AT LEAST I CAN HOUSE EVERYBODY
S OW W IH N D UW Y UW TH IH NG K W IY L N OW AH B AW T DH IY DH IY S EH K AH N AO F AH S ER AE T L IY S T AY K AH N HH AW S EH V R IY B
AA D IY

72(71):  1117130598.141 1117130599.55 transcript (01050926Z114645):
RIGHT
ALRIGHT
AO L R AY T

73(72):  1117130599.645 1117130605.922 transcript (01050926Z114647):
THAT IS THERE ANOTHER EQUIPMENT ROOM ON THE FLOOR THAT WE COULD MAYBE SHOOT ORANGES EQUIPMENT IN TWO
HAS IS THERE A ANOTHER EQUIPMENT ROOM ON THE FLOOR THAT WE COULD MAYBE SHOEHORN THIS EQUIPMENT INTO
HH AH Z IH Z DH EH R AH AH N AH DH ER IH K W IH P M AH N T R UW M AA N DH AH F L AO R DH AH T W IY K UH D M EY B IY SH UW HH AO R N DH
IH S IH K W IH P M AH N T IH N T AH

74(73):  1117130613.836 1117130616.588 transcript (01050926Z114710):
SO LET'S LOOK AT THE TIME_ALIGNED AREN'T SO
SO LET'S LOOK AT THE TIMELINE ALRIGHT SO
S OW L EH T S L UH K AE T DH AH T AY M L AY N AO L R AY T S OW

75(74):  1117130631.264 1117130633.293 transcript (01050926Z114720):
FOR A HUB OR LEAVE THAT
ALRIGHT I WELL I'LL LEAVE THAT
AO L R AY T AY W EH L AY L L IY V DH AE T

76(75):  1117130636.236 1117130640.598 transcript (01050926Z114727):
YES LET ME THAT TO YOU AND JOHN TO TO WORK OUT CLUES DETAILS
YEAH SO I'LL LEAVE THAT TO YOU AND JOHN TO TO WORK OUT THOSE DETAILS
Y AE S OW AY L L IY V DH AE T T IH Y UW AH N D JH AA N T UW T UW W ER K AW T DH OW Z D IH T EY L Z

77(76):  1117130640.598 1117130648.64 transcript (01050926Z114744):
I MIGHT MAKE IT MAY WE CAN PUT THE PRINTER IN INTO THE OTHER WITH THE OTHER PRINTERS AND FOR ROOM DOWN A WHOLE IN ANY THEN
THAT MIGHT MAKE IT AND MAYBE WE CAN PUT THE PRINTER IN INTO THE OTHER WITH THE OTHER PRINTERS IN THE PRINTER ROOM DOWN THE HALL IN ANY
EVENT
DH AE T M AY T M EY K IH T AH N D M EY B IY W IY K AH N P UH T DH IY P R IH N ER IH N IH N T AH DH IY AH DH ER W IH TH DH AH AH DH ER P
R IH N ER Z IH N DH AH P R IH N ER R UW M D AW N DH AH HH AO L IH N EH N IY IY V EH N T

78(77):  1117130648.255 1117130653.17 transcript (01050926Z114810):
WHICH TELLING ME IS WE'RE GONNA HAVE A ROOM FOR A JOB RUNNING A FEW DAYS AFTER
WHAT YOU'RE TELLING ME IS WE'RE GOING TO HAVE A ROOM FOR JOE_BROWNING A FEW DAYS AFTER
W AH T Y UW R T EH L IH NG M IY IH Z W ER G OW IH NG T IH HH AE V EY R UW M F R ER JH OW B R AW N IH NG AH F Y UW D EY Z AE F T ER

79(78):  1117130653.17 1117130654.484 transcript (01050926Z114824):
HE ARRIVES HERE WHICH
HE ARRIVES HERE WHICH
HH IY ER AY V Z HH IH R W IH CH
```

```
80(79):  1117130654.798 1117130657.464 transcript (01050926Z114828):
IS IS RIGHT UP ON THE LIMIT IS THAT CORRECT
IS IS RIGHT UP ON THE LIMIT IS THAT CORRECT
IH Z IH Z R AY T AH P AA N DH AH L IH M AH T IH Z DH AE T K ER EH K T

81(80):  1117130661.703 1117130666.217 transcript (01050926Z114837):
YEAH THE SECOND OFFICE THAT YOU'RE INTERESTED IN FIGHT FOR THE PRINTER AND DATA SERVER
NOW THE SECOND OFFICE THAT YOU HAD ORIGINALLY IDENTIFIED FOR THE PRINTER AND DATA SERVER
N AW DH AH S EH K AH N AO F AH S DH AH T Y UW HH AE D ER IH JH N AH L IY AY D EH N AH F AY D F ER DH AH P R IH N ER AH N D D EY T AH S
ER V ER

82(81):  1117130666.217 1117130667.226 transcript (01050926Z114853):
IS



83(82):  1117130667.341 1117130670.74 transcript (01050926Z114855):
THAT ALSO TO BE AVAILABLE THE SAME TIME
IS THAT ALSO GOING TO BE AVAILABLE AT THE SAME TIME
IH Z DH AE T AO L S OW G OW IH N T AH B IY AH V EY L AH B AH L AE T DH IY S EY M T AY M

== GESTURE ==============================================================
43:  1117130678.575 1117130682.263
[diamond(prob(6, 0.737)), writing(prob(12, 0.57), text(xx)), writing(prob(13, 0.53), text(1)), writing(prob(14, 0.51), text(I)), writing(prob(15,
0.5), text(N)), writing(prob(15, 0.49), text(Ni)), writing(prob(16, 0.48), text(Nr)), writing(prob(17, 0.46), text(Ns))]
=========================================================================

== GESTURE ==============================================================
44:  1117130683.247 1117130688.372
[writing(prob(14, 0.51), text(trails)), writing(prob(25, 0.312), text(h rails))]
=========================================================================

84(83):  1117130689.169 1117130691.788 transcript (01050926Z114905):
OKAY SO WHAT IS IT IS EIGHTY
OKAY SO IT IS IT IS A
OW K EY S OW IH T IH Z IH T IH Z EY

85(84):  1117130692.293 1117130693.512 transcript (01050926Z114913):

UH


86(85):  1117130693.684 1117130697.17 transcript (01050926Z114915):
IT'S A ACCEPTABLE OFFICE FOR PEOPLE IT'S NOT
IT'S AN ACCEPTABLE OFFICE FOR PEOPLE IT'S NOT
IH T S AH N AE K S EH P T AH B AH L AO F AH S F AO R P IY P AH L IH T S N AA T

87(86):  1117130697.17 1117130701.950 transcript (01050926Z114925):
THE PRINTER EQUIPMENT ROOM THAT WE'RE IT OR BRING PLOT THAT WE'RE TURNING INTO AN OFFICE
A AN EQUIPMENT ROOM THAT WE'RE GOING TO OR BROOM CLOSET WE'RE TURNING INTO AN OFFICE
EY AE N IH K W IH P M AH N T R UW M DH AH T W ER G OW IH NG T AH ER B R UW M K L AA Z AH T W ER T ER N IH NG IH N T AH AH N AO F AH S

88(87):  1117130704.445 1117130705.322 transcript (01050926Z114943):
ARE
ALRIGHT
AO L R AY T

89(88):  1117130711.569 1117130717.379 transcript (01050926Z114945):
SO THE WE'RE WE'RE WRITE_UP ON THE WIRE BUT IF EVERYTHING WORKS OUT WEEKEND
SO THAT WE'RE WE'RE RIGHT UP ON THE WIRE BUT IF UH EVERYTHING WORKS OUT WE CAN
S OW DH AE T W ER W ER R AY T AH P AA N DH IY W AY ER B AH T IH F EH V R IY TH IH NG W ER K S AW T W IY K AE N

90(89):  1117130717.703 1117130725.274 transcript (01050926Z115003):
GET THERE GET THIS NEW TEAM WORKING TOGETHER WITH MINOLTA LAYERS THAT WHAT YOU'RE YOU'RE SAYING IS THAT CORRECT
UH GET EVERY GET THIS NEW TEAM WORKING TOGETHER UH WITH MINIMAL DELAY THAT'S WHAT I HEAR YOU SAYING IS THAT CORRECT
```

```
G EH T EH V R IY G IH T DH IH S N Y UW T IY M W ER K IH NG T AH G EH DH ER W IH TH M IH N AH M AH L D IH L EY DH AE T S W AH T AY HH IH
R Y UW S EY IH NG IH Z DH AE T K ER EH K T

91(90):  1117130727.284 1117130729.655 transcript (01050926Z115027):
SO JACKED IN THE FINAL OBSERVATIONS
SO JACK ANY FINAL OBSERVATIONS
S OW JH AE K EH N IY F AY N AH L AA B Z ER V EY SH AH N Z

92(91):  1117130732.512 1117130734.17 transcript (01050926Z115033):
OKAY SURE
OKAY JOHN
OW K EY JH AA N

93(92):  1117130739.912 1117130741.264 transcript (01050926Z115037):
IS THAT A PROBLEM
IS THAT A PROBLEM
IH Z DH AE T EY P R AA B L AH M

94(93):  1117130751.112 1117130752.626 transcript (01050926Z115041):
LET'S GET
UH LET'S GET
L EH T S G IH T

95(94):  1117130752.626 1117130754.674 transcript (01050926Z115045):
GET THEM SITTING ON SOMETHING
GET THEM SITTING ON SOMETHING
G IH T DH EH M S IH T IH NG AO N S AH M TH IH NG

96(95):  1117130754.779 1117130757.836 transcript (01050926Z115051):
AND THEN WE'LL WORRY ABOUT MAKING IT LOOK NICE
AND THEN WE'LL WORRY ABOUT MAKING IT LOOK NICE
AE N D DH EH N W IH L W ER IY AH B AW T M EY K IH NG IH T L UH K N AY S

97(96):  1117130758.274 1117130762.722 transcript (01050926Z115101):
I MEAN HAVE TO ROLL EVERY YOU'RE WORKING WITH NINETEEN THIRTY SCREWS INFERRED RICHER HAVE
I MEAN AFTER ALL EVERYBODY HERE IS WORKING WITH NINETEEN THIRTIES PRISON FURNITURE ANYHOW
AY M IY N AE F T ER AO L EH V R IY B AA D IY HH IH R IH Z W ER K IH NG W IH DH N AY N T IY N TH ER T IY Z P R IH Z AH N F ER N IH CH ER
EH N IY HH AW

98(97):  1117130765.207 1117130767.141 transcript (01050926Z115116):
OKAY WELL THANKS VERY MUCH TILMAN
```

Biographical Note

Edward C. Kaiser was born in March, 1954 in Portland, Oregon, USA. He received his B.A. degree in American History from Reed College, Portland, Oregon in 1986, his A.S. in Software Engineering Technology from Portland Community College in 1996, and his M.S. degree in Computer Science from the Oregon Health & Science University, USA in 2005. He has worked previously as a research associate at the Center for Human-Computer Communication in the Department of Computer Science and Engineering at the Oregon Health & Science University (OHSU), and presently works as a research scientist for Adapx. Mr. Kaiser was a recipient of the 7th Annual OHSU Technology Innovation Award in recognition of his work on, "Dynamic Multimodal New Language Acquisition." Before joining Adapx, Mr. Kaiser has worked in OHSU's Center for Spoken Language Understanding (CSLU) and at SpeechWorks (now Nuance) in the area of natural language processing. His research experience and interests include spoken-language and multimodal systems, natural language processing, artificial intelligence, machine learning, and human-computer interfaces. Within the field of multimodal systems, Mr. Kaiser has worked and published extensively in the areas of hybrid/statistical multimodal integration, multimodal augmented and virtual reality environments, and speech and handwriting integration. At the 2006 8th International Conference on Multimodal Interfaces (ICMI '06) his paper, "Using Redundant Speech and Handwriting for Learning New Vocabulary and Understanding Abbreviations" received an OutStanding Paper Award. He has published more than twenty-five papers.