Rules for Automatic Mapping between Fast and Slow Speech

by

Stan Kwasny
Computer Science Department, Indiana University

Jonathan Dalby
Department of Linguistics, University of Edinburgh

Robert Port
Department of Linguistics, Indiana University

TECHNICAL REPORT NO. 175

by
Stan Kwasny
Computer Science Department, Indiana University

Jonathan Dalby
Department of Linguistics, University of Edinburgh

Robert Port
Department of Linguistics, Indiana University
July, 1985

Rules for Automatic Mapping between Fast and Slow Speech

by

Stan Kwasny
Computer Science Department, Indiana University

Jonathan Dalby
Department of Linguistics, University of Edinburgh

Robert Port
Department of Linguistics, Indiana University

TECHNICAL REPORT NO. 175

by

Stan Kwasny
Computer Science Department, Indiana University

Jonathan Dalby
Department of Linguistics, University of Edinburgh

Robert Port
Department of Linguistics, Indiana University

July, 1985

Rules for Automatic Mapping between Fast and Slow Speech[a]

Stan Kwasny

Computer Science Department, Indiana University

Jonathan Dalby

Department of Linguistics, University of Edinburgh

Robert Port

Department of Linguistics, Indiana University

1. Introduction

This paper reports preliminary results of a project to deal with one aspect of the enormous phonetic variability of language. In speech production a great many rules apply that modify – sometimes grossly – the pronunciation of words. All of us are familiar with such typical pronunciations as "Jeet yet?" for "Did you eat yet?" Such massive destruction of phonological material is not unusual in rapid speech of normal speakers.

The difficulty this raises for speech recognition is that a phonetic transcription would be not be very helpful for finding a word in the dictionary. One might propose that linguists simply do phonological research on the problem to come up with a set of phonological rules sufficient to describe these patterns. Several attempts have already been made along this line (Oshika, et al, 1975; Dalby, 1984; Bailey, 1980, Zwicky, 1972). These rules could then be employed as part of the speech recognition system that is capable of generating a fairly reliable segmental phonetic transcription of speech to "preprocess" the transcription by inverting the effects of the rules before words are looked up in the dictionary.

Still there would be a major problem in our view. Speakers apparently economize on their speech effort in very different ways. Thus it is unlikely that linguists could achieve a satisfactory description of this relationship. Such a large number of speaker-idiosyncratic rules apply that discovering the rules by normal linguistic procedures would become extremely laborious. In such a situation, an intelligent system capable of discovering the phonological rules which describe the speaking patterns of individual speakers given transcriptions of utterances at two speaking tempos would be very useful. These rules would permit the recognition system to be automatically tuned for tempo variations of each speaker.

---

[a] This report is an expanded version of a paper of the same title which appeared in *Research in Phonetics 4*, June, 1984, Bloomington, IN, 235-248. It is based on a presentation delivered to the Kentucky Foreign Language Conference, April, 1984.

This stage of the project involved an investigation of the extent of the problem of variation in pronunciation during rapid speech and initial steps toward discovery of the phonological rules implicit in our data base. As a first step toward rule discovery, we developed a program that matches phonetic transcriptions of fast and slow sentences in our data in order to discover the contextual characteristics of those deletions. We are in the process of developing a program to discover these contexts.

Thus, our overall goal is to construct a system that would be able to find phonological rules for spoken English that would permit recovery of the standard dictionary entries for the words. We ultimately expect to be able to do this entirely automatically given phonetic transcriptions.

Phonetic Alphabet. The first step in this project was to develop a scheme for phonetic transcription that could be easily read by a linguist yet typed on a computer and read by Lisp programs. Many years ago the ARPAbet was developed as a machine-readable phonetic alphabet, but it was based on 6-bit ASCII code, which features only upper case and is difficult to read. We developed our own alphabet, which is similar to these, and present a sample transcription in Figure 1.

[dh a . p' I k el z . ah r . t' u . s au er]
"The pickles are too sour."

Figure 1

In order to permit atomic symbols using one, two, or three ASCII characters in both upper and lower case, we employed the space character as a separator. This facilitates the representation of transcripts within our Lisp programs. We also use the period ('.') as a word boundary marker. The other symbols were chosen to be similar to IPA or other symbols commonly used for the phonetic alphabet. The particular advantage of this system is that it permits indefinite expansion of the symbol set as needed. See Appendix A for the complete alphabet.

2. Speech Production Experiment

An experiment on speech production was conducted in order to discover the contexts in which vowels are deleted. Utterance transcriptions were prepared manually by carefully reviewing the tapes and inspecting the spectrogram display while the analysis was performed automatically.

2.1. Methods

A set of 182 sentences was prepared containing a large number of schwa vowels and read at both slow and fast tempos by three speakers. Transcriptions were made of these utterances for use in the rule discovery portion of the project.

Materials. The sentences used in the experiment were constructed so as to contain as many unstressed vowel locations in as many consonantal environments as possible and still yield reasonably plausible utterances. An attempt was also made to use many different sentence types and semantic contexts in order to stave off the boredom inherent in the task of reading such lists and thus to prevent the rhythmic regularity that often characterizes the reading of carrier sentences. Indeed it was hoped that the variety and light-heartedness of some of the sentences would help the subjects relax. For the most part, subjects had little difficulty in reading the sentences fluently.

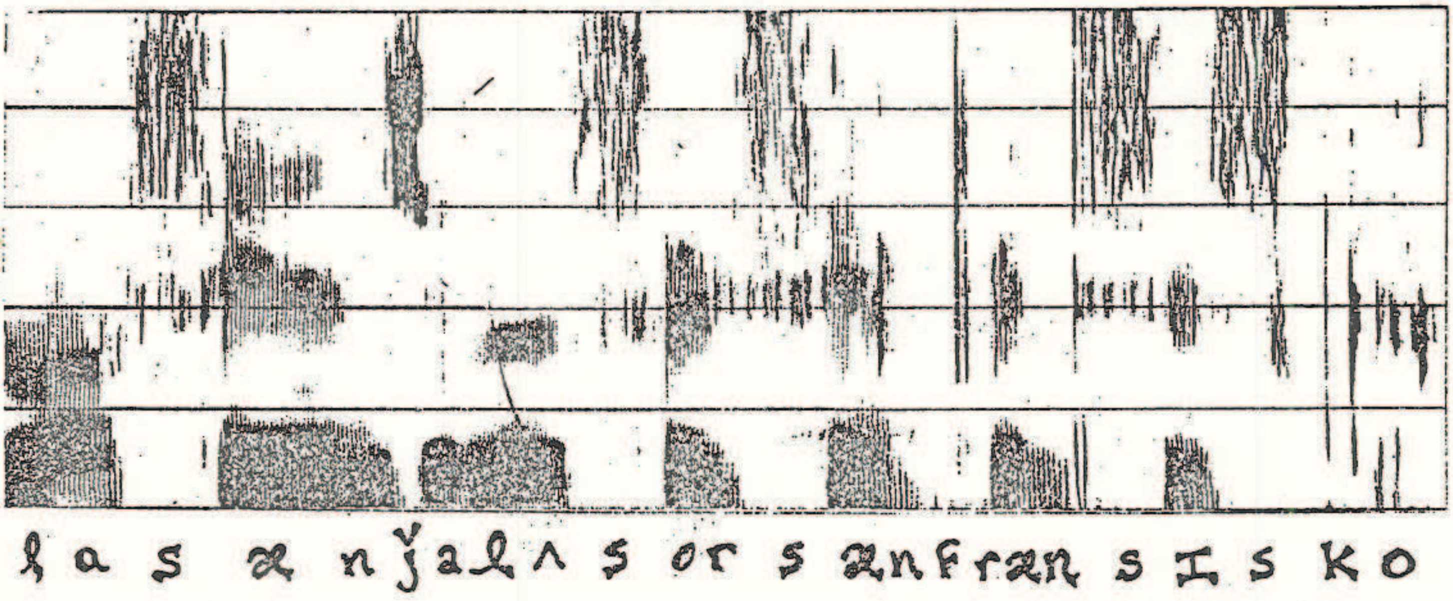
The 182 sentences were divided into 18 lists of about 10 sentences each. They contained a total of about 900 unstressed vowels that could, in principle, be deleted. See Appendix B for a complete list of sentences.

Subjects. Since it is difficult to elicit relatively unmonitored speech in a formal experimental environment, we chose to use subjects who were familiar with the laboratory environment and who would not be too self-conscious about their speech patterns. Therefore two graduate students from the Department of Linguistics and one graduate student from the Department of Folklore were chosen as subjects. All three were well known to the experimenter and were familiar with such tasks as well as with the Phonetics Laboratory. The subjects were told that they were participating in an experiment on fast speech and were asked to read the 18 sets of sentences first at a normal reading tempo and then as fast as possible without stumbling. If after any sentence the subject felt that he had made an error or that an utterance wasn't really an acceptable example of an utterance he might produce naturally, he/she was encouraged to repeat the sentence until satisfied with it.

Procedure. One of the experimenters (J. Dalby) listened to each recorded sentence and transcribed it in our own version of the IPA phonetic alphabet. Since the decision as to whether or not some sort of unstressed vowel was present in an utterance or not is a difficult one, broad-band spectrograms were made to assist in the process of transcribing the utterances. For example, Figure 2 shows a fragment of a sentence at both tempos by one of the speakers. It is easy to see how many fewer syllables there are in the fast sentence than in the slow. Not only are a number of vowels deleted, but several other major assimilations also occur. Note that "San Francisco" becomes [s ae m f s I s k o]. Using the two channels of information it was possible to bring the transcription process to a high degree of accuracy. This set of transcriptions served as the data base for the analysis portion of the project.

3. Comparing Fast and Slow Transcriptions

Analysis of the transcription sentence pairs consists of a two-step process. In the first step, two transcriptions representing the slow and fast utterances of the subject are aligned according to a heuristic "brute-force" *matching algorithm* coded in Lisp. The purpose is to determine which vowels and consonants in the two transcriptions correspond to each other. This is tricky to do automatically since generally there are a different number of segments in the two transcriptions and different segmental units may be employed. For a good overview of some of the computational aspects of this problem, see Sankoff and Kruskal (1983).



"... Los Angeles or San Francisco"

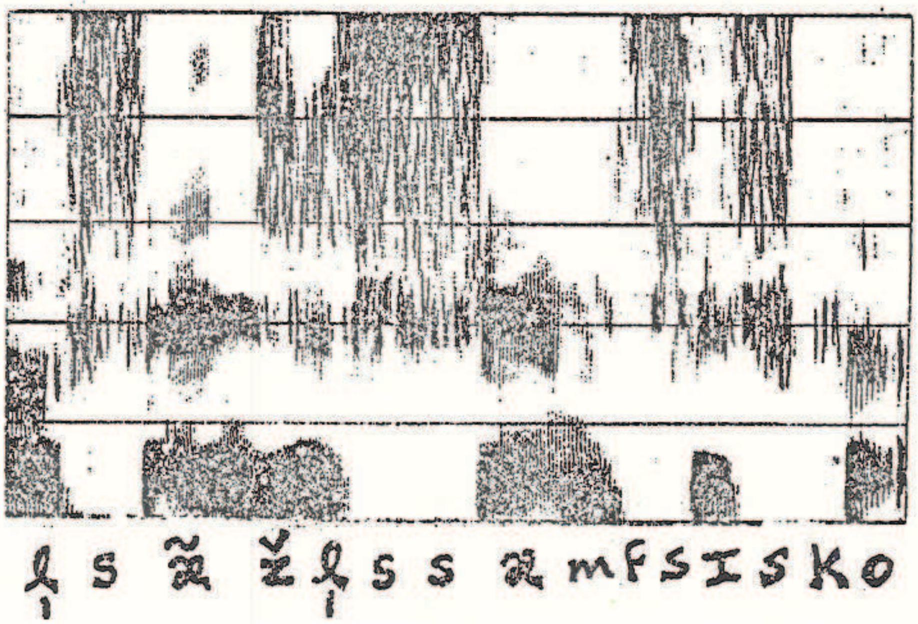


Figure 2
Spectrogram of Sentence Fragment

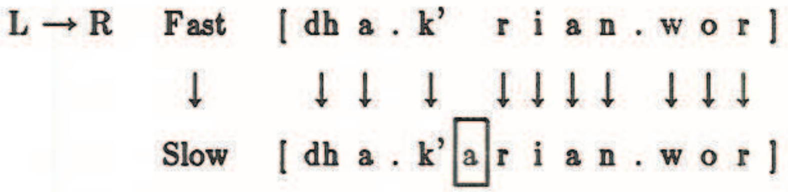
At the conclusion of this first step, the original matching results are analyzed. Statistics on the matching process are gathered in the form of a table which shows segmental matches, insertions, deletions, and changes. Appendix C contains an abridged[b] form of that table. It shows the relative ordering of segments based on the percentage of matches for that segment. We shall call this measure the *stability* of the segment.

In the second step, the original matching results along with the analysis results are used by the recursive *re-matching algorithm* to refine the matching process. At the end of this step, the results are again analyzed to collect new statistics on the effectiveness of the refined matching. These results are also tabulated in Appendix C in order to invite comparisons with the previous table. At present, the matching process works pretty well. We are in the process of implementing the rule discovery procedure.

3.1. Development of the Matching Algorithm.

The two transcripts, representing the slow utterance and the fast utterance, are compared in the brute-force matching algorithm. Our goal in designing the algorithm was to align the tokens of the transcripts as best as possible in the absence of any real phonetic knowledge.

Figure 3 illustrates some of the problems to be dealt with in doing the matching of the two transcriptions. The first problem concerns the number of segments to be matched. Since the Slow transcription typically has more segments in it than the Fast one, we want the matching algorithm to follow the Fast one and, if the next symbol doesn't match, to search beyond the immediately adjacent symbol to seek a match. Thus in the figure, the missing schwa in "Korean" is handled correctly as a missing segment.


L → R
Fast
[dh a . k' r i a n . w o r]
↓
Slow
[dh a . k' a r i a n . w o r]

The Korean war ...

Figure 3[c]

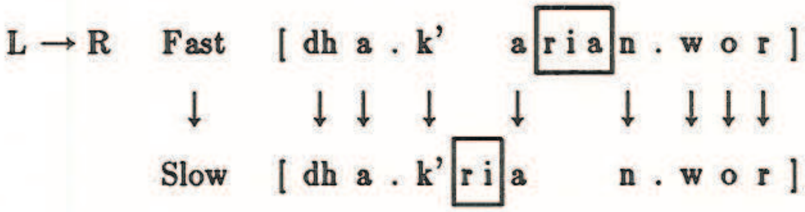
Our brute-force matching program will look as many as five tokens ahead to find

---

[b] We have shortened the table by omitting those segments which do not appear at least ten times in the slow transcriptions. We also do not show statistics on insertions, deletions, and changes.

[c] The direction of the matching is shown from left-to-right and the segments of the fast transcription are matched to segments of the slow transcription. Unmatched segments are boxed. We will use these conventions throughout the remainder of the paper.

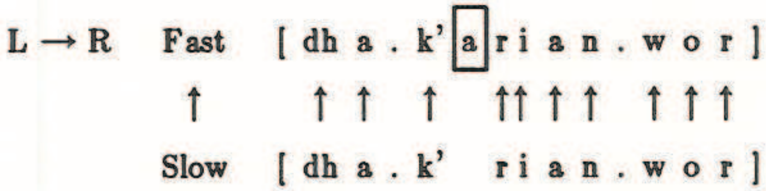
a match. This implies that up to four contiguous segments may be deleted without jeopardizing the match process. Unfortunately, sometimes a segment appears in the Fast form that is missing from the Slow, citation-like form. If that segment recurs coincidentally within the five-symbol window, then a misalignment will occur. For example, if it were the first schwa in "Korean" that were missing from the Slow transcription, then a major mismatch will occur, as shown in figure 4.


L → R
Fast [dh a . k' a r i a n . w o r]
Slow [dh a . k' r i a n . w o r]

The Korean war ...

Figure 4

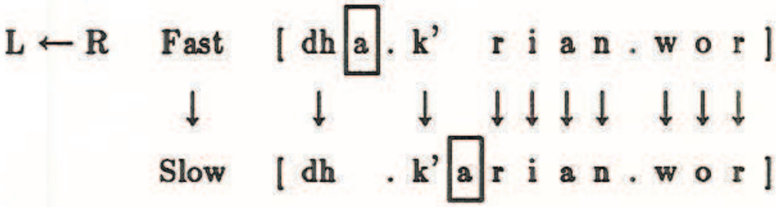
This error can be caught, however, if we match slow-to-fast rather than fast-to-slow as shown in Figure 5.


L → R
Fast [dh a . k' a r i a n . w o r]
Slow [dh a . k' r i a n . w o r]

The Korean war ...

Figure 5

Finally, there are cases which require that we perform matching right-to-left for the correct alignment of segments. This is illustrated by figure 6.


L ← R
Fast [dh a . k' r i a n . w o r]
Slow [dh . k' a r i a n . w o r]

The Korean war ...

Figure 6

If matching were conducted left-to-right, then both fast-to-slow and slow-to-fast strategies would cause a mismatch of the schwas. The only correct matching

```
1. For each of the two directions
                    { Left-Right, Right-Left }
do

    2. For each of the two comparison orders
                         { Slow-Fast, Fast-Slow }
    do

        3. Consider the first segment in the first transcription.
        Attempt to exactly match that segment
            with a segment in the second.
        Look up to five segments ahead in the second.

        4. If step 3 fails to find a match,
            then repeat step 3,
                     but attempt to match segments
                     within one feature of each other.

        5. If no match has been found,
            then the segment under consideration has been deleted
            else the segment is aligned with its matched segment.

6. After all four segment matches are complete,
    select the one which contains the fewest unmatched segments.
```

"Brute-Force" Matching Algorithm
Figure 7

strategy in this case is right-to-left, slow-to-fast. For this reason, we do the matches in *both* directions – from left-to-right and from right-to-left.

Thus, we count the number of unmatched symbols (the ones shown in the boxes), and select the list that has the fewest unmatched segments. Unfortunately, this is still not sufficient. Since it turns out that Slow speech does not always contains all the deletable characters, we must do the matching in all four combinations of Slow-to-Fast/Fast-to-Slow and Left-Right/Right-Left in order to be sure not to miss any symbols.

There is one more problem we attempted to account for in our data. Frequently, a segment match is missed because the segment has changed its phonetic features in some way. For example, frequently a vowel is nasalized or made more lenis, as from [i] to [ɪ]. For this reason we implemented a system to check whether a segment for which no match could be found differed from one of the candidates by a single feature. Figure 7 contains a statement of the algorithm discussed above.

With this mixture of heuristics, the matching algorithm seems to perform well enough to accomplish our goal of providing the initial data concerning deletability of segments. It is interesting to examine the behavior of the algorithm. Results of a frequency count of the use of the different branches in the algorithm is shown in Figure 8. The numbers do not sum to 100% because we counted those matches which were ties twice. The test for a match differing in a single feature was successfully employed in about 14% of the cases.

	L→R	R→L
Fast→Slow	55%	50%
Slow→Fast	13%	13%

Frequency of Use of Heuristics
Figure 8

The algorithm performed quite well on most of the data. It can be seen that, as expected, matching from fast to slow works much better than from slow to fast. There is not much difference overall between left-to-right or right-to-left matching. In particular cases, however, it obviously makes a big difference. The most difficult problems for the brute-force algorithm occurs in those pairs of transcriptions which contained a problem spot both at the beginning which caused the left-to-right strategies to fail and toward the end which caused the right-to-left strategies to fail. In such cases, neither strategy is clearly superior.

Sometimes the matcher works remarkably well according to one's intuitions. For example, Figure 9 shows how it performed on the phrase "Los Angeles or San Francisco" displayed above.

```
[ l a s ae n   jh a l uh s . or . s ae n f r ae n s I s k o ]
  |   | |           |    |        | |    |          | | | |
[ el  s ae~    zh   el   s      . s ae m f        s I s k o ]
```

". . . Los Angeles or San Francisco."

Figure 9

It can be seen that the segments were lined up very well with the exception of the missed correspondence between "jh" and "zh". These two segments differ by two features in our feature system: [delayed release] and [continuant].

Subsequent analysis of the matched transcripts reveals a few remaining errors. Here is a good example.

```
[ k r E p t . uh p . ah p . ah n . dh E m . g r ae zh u l i . ]
  | | | | |                    |
[ k r E p t . a m . uh m .     n      E m . g r ae jh a l i . ]
```

". . . crept up on them gradually."

Figure 10

This is the last portion of a sentence matched from left-to-right. The drastic phonetic changes in pronunciation – from "p" to "m" and the assimilation from "dh" to "n" – which all occur near each other threw the matcher off track so badly that it was not able to recover for the final word *gradually* which should have been easy. Obviously, if matched from right-to-left, it would handle *gradually* just fine, but then the earlier portion of the sentence would be thrown off.

Performance of Brute-Force Matcher. The overall performance of this matching algorithm is surprisingly good. For our three speakers, the percent of slow-speech segments that were judged to be adequately matched averaged about 92%. The percent of matches for the individual speakers ranged from 91% to 94%. Looking at complete sentences, 24% were judged to contain no errors at all and about 2/3 of the sentences contained 2 or fewer segmental errors. Although this performance is quite good, we felt that it was still not good enough for our purposes. The kind of problems exhibited just above reduced the effectiveness of the matcher.

3.2. Extensions to the Matching Algorithm.

Such problems can be corrected if a rather different approach is taken in which "islands of reliability" are used to guide the matching process. From the data, it is clear that some segments are much less likely to be deleted than others. We can prevent the kind of gross mismatches, such as we just examined, if we match the most reliable (i.e., "stable") segments on the first pass. For example, stressed or full vowels are highly resistant to deletion or even to significant change in feature content (Dalby, 1984). Only after these segments have been located will other matches be attempted.

Of course, selection of such "reliable" segments could be performed *a priori*, but this would be error-prone and not responsive to the individual differences present in the data. Instead, the frequency counts from the data obtained in the brute force matching algorithm are used to determine an ordering among segments (see Appendix C). Although the brute-force matching algorithm clearly makes errors, it proved to be good enough to provide a basis for ordering segment types according to the percentage of successful matches. Figure 11 contains a description of the re-matching algorithm.

From the algorithm, you can see that the data collected from the brute-force matcher determines the ordering of the segment list. This ordering determines, in the grossest way possible, where to commit to particular alignments. In those cases where good clues for matching do not exist, the original matching algorithm is again utilized to fill in the gaps between the reliable "islands." In this way, the re-match algorithm is guaranteed to do no worse than the brute-force algorithm in matching substrings of segments which it finds difficult. Such difficulties surface in the re-match algorithm in a way that permits it to avoid making dramatic errors. Finally, we compare the re-matched transcripts with the originally matched ones and select the alignment which contains the fewest unmatched segments.

Because we are utilizing results from the initial algorithm, we are actually "bootstrapping" the matching process from earlier results to make a better matcher. This bootstrapping activity could, of course, continue with the data collected from the re-matching algorithm determining an ordering which could again drive the re-matcher. As long as matching performance continued to increase, it might be worth pursuing this possibility.

Performance of Re-Matcher. The alignments produced by the Re-Matching algorithm were compared with those of the original brute-force matcher. The result that contained the most alignments was judged to be the better match. The structure of the Lisp representation for alignments permitted as simple length test to make this judgement. In doing these comparisons, there are four possible, mutually exclusive outcomes. Figure 12 contains the results.

1. The two matchings were identical (tie);
2. The two matchings did not match identically, but were identical in length (length-tie);
3. The original match gave more alignments than the re-match (brute-force matcher wins);
4. The re-match gave more alignments than the original match (re-matcher wins).

```
1. Given a list of segments ordered by stability and
      two transcripts to match:

2. If the most stable segment occurs an equal number of
      non-zero times in each of the two transcripts
   then
      align those corresponding segments and
      recursively attempt to match between them
   else
      eliminate the most stable segment from the ordered list and
      recursively attempt the match again.

3. If the list of stable segments becomes empty,
      matching will be impossible due to an unequal number
      of stable segments in the two transcripts.
      If this happens, we use the brute-force matching algorithm.
```

Re-Matching Algorithm
Figure 11

Outcomes	Freq	%
Exact matches	247	45
Length ties	40	7
Original match	18	3
Re-Match	241	44
Totals	546	99

Results of Re-Match
Figure 12

Length Difference	Brute-Force Matcher	Re-Matcher
1	9	47
2	1	23
3	5	28
4	1	14
5	1	15
6	0	12
7	1	13
8	0	14
9	0	8
10	0	18
11	0	10
12	0	6
13	0	9
14	0	5
15	0	5
16	0	3
17	0	4
18	0	3
19	0	2
20	0	1
21	0	1
	18	241

Results of Re-Match
Figure 13

In the latter two categories, we also counted what the difference in lengths were between the two matchings to get an idea how much additional matching was going on. That is, whenever the original matcher got a better result, we measured how much better that result was by looking at the difference in lengths between the two pairs of matches. We did likewise whenever the re-matching got a better result. Those results are shown in Figure 13. Clearly, 89% of the time the re-matching process tied or bettered the original algorithm (241+ 247). We also see good evidence that significantly better matching occurred in many cases. On average, this calculates out to a gain in the first case of 1.20 (42/35) matches better, while in the second case we get an average gain of 6.34 (1528/241) matches better.

Another way to consider these figures would be to calculate the average amount of shift in number of matches. We do this by considering the original matches as negatively-weighted amounts, the re-matches as positively-weighted amounts, and the exact matches and ties as neutral (zero) shifts in matching. Doing this, we get 2.72 ((-42 + 1528)/546) as the average shift in matches in the direction of the re-match algorithm. This says that on average, the re-matcher will find 2.72 more matches between the two transcriptions than the brute-force matcher. Notice that this is heavily influenced by the number of exact matches and ties that have occurred. Furthermore, since the original brute-force matcher was judged to perform fairly well, this seems a remarkable result. Also, since the original matcher now only performs better on 18 (3%) of the samples, the re-matching process does at least as well as the original matcher in about 97% of all samples. Appendix D contains sample output of the combined matching procedure.

4. Conclusions.

Although we are clearly in the early stages of this project with regard to automatically extracting rules from data, our results seem to suggest several interesting conclusions.

1) First, it is possible to use relatively raw linguistic input and to process it in a way that permits useful linguistic generalizations to be drawn from the data. We believe the entire rule-extraction process can be done without direct intervention by a linguist once the system is developed.

2) This analysis of phonetic data may prove useful in automatic speech recognition if we assume that acoustic analysis can yield a high-quality phonetic transcription. Since it is likely that individual speakers have rather different rules for rapid speech or for other kinds speech reduction, an automated system that can customize a speech recognizer to the idiosyncrasies of individuals would be highly valuable.

3) We have demonstrated an approach to matching phonetic transcriptions that is a bootstrapping procedure and that can improve the accuracy of its performance iteratively. Even though the results of our relatively ignorant brute-force algorithm were judged to be quite good, this bootstrapping process managed to improve that original algorithm significantly.

5. Future Work.

We plan to use the high-quality matching algorithm that we have developed as a basis for rule extraction. We propose to pursue this as a statistically-based procedure wherein we keep track of the frequency of various contexts for deletions, insertions, and changes. In this approach, the credibility of a hypothesized rule will be increased according to how often evidence is found for that rule in the data. A rule will be hypothesized based on the number of times it is observed in the data relative to the number of possible times a rule might have applied.

Refinement of a rule (e.g., noting exceptions) will be performed using the "example" and "near-miss" approach suggested by Winston (1975). Without human intervention, however, the program will not actually know if a given sample is an example or a near-miss, but could conceivably pursue both paths, ruling out the least plausible at some later point. Michalski and Stepp (1983) have developed a procedure called CLUSTER, which could perhaps prove to be a more promising tact.

6. References

Bailey, Charles-James M. (1978) *Gradience in English Syllablization and a Revised Concept of Unmarked Syllablization.* Indiana University Linguistics Club, Bloomington, IN

Dalby, J.(1984) *The Phonetic Structure of Fast Speech.* Unpublished Ph.D. dissertation, Indiana University.

Michalski, Ryszard S., and Robert E. Stepp (1983), "Learning from Observation: Conceptual Clustering," in Ryszard S. Michalski , Jaime G. Carbonell, and Tom M. Mitchell (eds.) *Machine Learning —An Artificial Intelligence Approach,* Tioga Publishing Company, Palo Alto, CA.

Oshika, B., V. Zue, R. Weeks, H. Nuy, J. Aurbach (1975), "The Role of Phonological Rules in Speech Understanding Research". IEEE Trans on Acous, Spch and Sign Proc. ASSP-23, 104-112.

Sankoff, David and Joseph Kruskal (1983). *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison.* (Addison-Wesley; Reading, Mass).

Winston, Patrick Henry (1975) "Learning Structural Descriptions from Examples". PhD thesis, in Patrick Henry Winston (ed.) *The Psychology of Computer Vision,* McGraw-Hill Book Co., New York.

Zwicky, Arnold (1972) "Note on a phonological hierarchy in English". Stockwell, Robert P. Macaulay, Ronald K.S. (eds.) *Linguistic Change and Generative Theory,* Indiana University Press, Bloomington and London, 275-301.

Appendix A

Phonetic Alphabet in ASCII

VOWELS	as in	CONSONANTS	as in
i	*beet*	p	IPA
I	*bit*	t	IPA
e	IPA	k	IPA
E	*bet*	b	IPA
ae	*bat*	d	IPA
aa	Boston broad a	g	IPA
ah	*hot* (low mid a)	f	IPA
ao	(low back a)	th	IPA theta
oa	*bought* (open o)	s	IPA
o	IPA	sh	*show*
U	*put* (IPA)	kh	IPA x
u	IPA	v	IPA
a	‘schwa’	dh	'eth'
uh	*but* (IPA ^)	z	IPA
ih	*roses* (front schwa)	zh	z wedge
er	*bird* or *butter*	gh	IPA gamma
DIPTHONGS		ch	c wedge
au	*house*	jh	j wedge
ou	*boat*	m	IPA
oi	*boy*	n	IPA
ai	*why*	ny	palatal n
ei	*day*	ng	'eng'
DIACRITICS		y	IPA j
C:, V:	long segment	w	IPA
Ṽ	nasalized vowel	r	red
C'	aspirated stop	l	*led*
C?	glottalization	h	IPA
eC	syllabic C	D	apical flap
C%, V%	devoiced segment	q	glottal stop
C&, V&	voiced segment		
C+	fortis		
C-	lenis		
'C	velar plosion		

Appendix B

Test Sentences of the Fast Speech Study

1. This tight collar is killing me.
2. In my opinion Picasso is a genius.
3. Al's physician suggested a lobotomy.
4. This journalist has a marvelously limited vocabulary.
5. Can you arrange to be here before eleven?
6. My folks live in Missouri.
7. I find Carl insensitive and insufferably prejudiced.
8. In an emergency it is important to remain calm.
9. I like marigolds but I can't stand petunias.
10. Lorraine's psychiatrist thinks her neurosis is improving.
11. Just focus the camera and shoot.
12. Phelps was arrested for felonious assault.
13. I find geometry more difficult than algebra.
14. Telegraphers must know Morse Code perfectly.
15. How far is Topeka from Kansas City?
16. The only thing Clyde ever reads is the gossip column.
17. Seven Bolivian Villages were devastated in the catastrophe.
18. Brian is usually excessively polite.
19. Only immediate family are invited to the marriage ceremony.
20. This common principle defies rigorous definition.
21. A hammer and chisel will break this cement.
22. Aristotle was a Greek philosopher.
23. Pakistan has developed the atomic bomb.
24. Naturally the winner was exhilarated.
25. It is easier to multiply than to divide.
26. Florida is an example of a peninsula.
27. He has probably bragged about it before.
28. The legislature has adopted the opposite position.
29. Was Joseph Kennedy an Irish immigrant?
30. Messages were sent but no communication occurred.
31. Who won custody of the children?
32. The error went undetected at the embassy.
33. The fidelity of the amplification left a lot to be desired.
34. South Dakota is no paradise in December.
35. A minute ago you said you were positive.
36. Is Eric's trained gorilla named Napoleon?
37. I think Karen's humility is just a facade.
38. Is Pomona near Los Angeles or San Francisco?
39. Mechanical engineers must be familiar with mathematics.
40. The telephone company is the country's biggest monopoly.
41. Phyllis thinks my new orange linoleum is horrid.
42. Government authorities anticipate another fuel shortage.
43. Vanilla flavoring is derived from a bean.
44. Carolyn is a horrible pessimist.
45. Elizabeth is in the hospital for minor surgery.
46. The Chicago Symphony's Vivaldi festival begins tomorrow.
47. How far is Buffalo from Detroit?
48. The photography in the film was excellent.

49. His irrational jealousy exasperates her.
50. Only an imbecile would attempt to justify despotism.
51. The debate in the Senate begins today.
52. Rush-hour traffic is a pedestrian's nightmare.
53. Jupiter is the largest planet in the galaxy.
54. Upward mobility characterizes suburbia today.
55. This Mexican tequila is liquid dynamite.
56. My favorite dessert is lemon meringue pie.
57. Take the elevator up and the escalator down.
58. Will the therapy reduce John's bizarre behavior?
59. Nevada is famous for casinos.
60. Their rigid attitudes surprise me.
61. Alaska is famous for its rugged terrain.
62. The English Channel is about thirty kilometers across.
63. Bavarians think American tourists are extravagant.
64. Carole's cousin LaMont died of pneumonia.
65. Warren demanded an apology from the sheriff.
66. The Defense Department always favors increased military spending.
67. It is a mistake to encourage this publicity.
68. Will the President hold the line on deficit spending?
69. Nobody I know has ever lived in Arizona.
70. What's the difference between the numerator and the denominator?
71. The Hawaiian Islands are in the Pacific Ocean.
72. Juvenile delinquents are committing more and more felonies.
73. The owl's attack on the rabbit was savage and swift.
74. I wouldn't subscribe to that magazine for a million bucks.
75. Is Omaha the capital of Nebraska?
76. Morris Halle is a famous phonologist.
77. Female lions are generally more ferocious than males.
78. The prosecution lacks sufficient evidence.
79. There is no limit to Ann's capacity for optimism.
80. I believe camels are extremely intelligent animals.
81. Federal tobacco subsidies should be eliminated.
82. Renae thinks religious discussions are illogical.
83. I don't know what instruments Laverne plays besides guitar.
84. The escaped prisoner is armed and dangerous.
85. The victory celebration lasted the entire night.
86. Alex was decorated for bravery in the Korean War.
87. The Mississippi is longer than the Amazon.
88. Did you say necessary evil or evil necessity?
89. Unfortunately propaganda's not always easy to recognize.
90. I suspect Carl won't support that proposal.
91. Senility and death are hereditary.
92. I never miss the Fourth of July parade.
93. Malaria is a tropical disease.
94. I didn't think MacMillan was capable of belligerence.
95. Morocco used to be a French colony.
96. The baritone saxophone is a cross between a cannon and a kazoo.
97. Gasoline is a dollar forty a gallon.
98. Florida is famous for alligators.
99. Factory work is extremely monotonous.
100. Gambling losses aren't tax deductible in Nebraska.
101. Edison's physical condition is wretched.
102. Your devious purpose is obvious.

103. I have never met an Arab with a harem.
104. Is there a difference between a waltz and a gavotte?
105. The judge has achieved a degree of material success.
106. Decapitation for petty larceny seems a bit severe.
107. The architecture of this dormitory resembles Alcatraz.
108. That service station attendant looks cadaverous.
109. The War Memorial has been designated a National Monument.
110. The downtown vicinity is gradually losing its vitality.
111. The pirate's final destination was the Solomon Islands.
112. The cable is corroded where the acid has leaked.
113. Try not to collide with the garbage can.
114. One potato, two potato, three potato, four...
115. That service station tends to run out of regular.
116. The administration's proposal is considered inadequate.
117. You grill a burger while I fetch the catsup.
118. Expectations of expectoration prompted installation of spittoons.
119. Negotiators finally agreed to an armistice.
120. Dense is not the word.
121. Lewis is growing begonias in Oregon.
122. Zabrinsky was sent to expedite the investigation.
123. What's the difference between a chrysalis and a cocoon?
124. I'll take a pistachio sandwich and hot pastrami ice cream.
125. Latitude and longitude are necessary for navigation.
126. We'll set up the pavilion on the veranda.
127. Intoxication crept up on them gradually.
128. The engine coughs out clouds of black smoke.
129. The objections of the treasurer were almost inaudible.
130. This is an extraordinarily delicious banana.
131. The lame minister limped to the pulpit and prayed.
132. Plant carrots and tomatoes early this year.
133. Children should be seen and not heard.
134. He used to sport a chrysanthemum in his lapel.
135. Cassidy's personal life is stupefying.
136. His grades in college were way below average.
137. The cabinet is anxious to find a practical solution.
138. The Constable testified that the wallets were stolen.
139. "Accuracy is not optional," the editor emphasized.
140. He has the resolution typical of a dedicated socialist.
141. Katherine's self-confidence is unshakable.
142. The consequences of the abdication were significant.
143. We listened to the dilapidated organ gasp and groan.
144. Her scholastic achievements are spectacular.
145. This thesis is based on a corpus of ballads.
146. I warn you not to succumb to unreasonable demands.
147. Only a jackass would bray like that.
148. Old glory received the regiment's salute.
149. The broken tennis racquet will be replaced.
150. Ramifications of this arbitrary decision are numerous.
151. The village idiot carts the ashes away.
152. The union refused to submit to binding arbitration.
153. His losing the precious medallion was impossibly irresponsible.
154. This seminar is as crowded as Noah's ark.
155. There is scum in the bilge of the sloop.
156. This jewel has many facets or faces.

157. There are some oak trees that benefit from little rain.
158. Their hospitality included a stupendous spaghetti dinner.
159. A kiln is an oven to fire ceramics in.
160. The escapade turned out to be totally fictitious.
161. I've wracked my brain for the solution.
162. President Carter was livid with rage.
163. The bandit who robbed them was ragged and bald.
164. Since his divorce, Phillip's plight is pathetic.
165. Professional organizations serve several purposes.
166. Surely an occasional drink can do no harm.
167. The Canadian dentist's office is in Toronto.
168. The accompanist was subsequently shot.
169. Wait for the crisis to blow over.
170. Loose is what the Elizabethans were.
171. Lorenzo's uncle is a bishop in the Episcopal Church.
172. A horde of serfs stationed themselves around the carriage.
173. Alice cultivated a purposeful anonymity.
174. The banded flamingos have all flown away.
175. The new recruits lived in decrepit barracks.
176. Hillary succeeded where others had failed.
177. The bite of a rabid animal is dangerous.
178. Believe it or not they finally found the corpse in the caboose.
179. The priest of this parish always fasts on the Sabbath.
180. Steven managed an approximation of sobriety.
181. Our fresh pastries have calories galore.
182. Her parents expect her to join the green berets.

Appendix C
Segment Ordering in Matching Algorithms
(Ordered by Stability)

Segment	Rank in Matching Algorithm	Matches / Total (Stability)	Rank in Re-Matching Algorithm	Matches / Total (Stability)
ei	1.	0.812	2.	0.92
ch	2.	0.796	13.	0.833
f	3.	0.79	3.	0.908
s	4.	0.788	4.	0.894
ah	5.	0.784	5.	0.89
m	6.	0.763	6.	0.889
sh	7.	0.755	1.	0.922
u	8.	0.754	8.	0.854
r	9.	0.751	12.	0.846
y	10.	0.747	9.	0.852
l	11.	0.742	10.	0.85
o	12.	0.74	11.	0.848
E	13.	0.74	7.	0.888
k	14.	0.707	14.	0.824
b	15.	0.7	15.	0.803
g	16.	0.686	17.	0.781
ng	17.	0.686	20.	0.761
i	18.	0.685	16.	0.789
ae	19.	0.683	22.	0.75
au	20.	0.673	23.	0.734
p	21.	0.67	18.	0.78
w	22.	0.669	21.	0.751
er	23.	0.652	25.	0.727
uh	24.	0.643	19.	0.762
jh	25.	0.642	26.	0.727
th	26.	0.629	24.	0.728
ai	27.	0.623	27.	0.706
z	28.	0.602	30.	0.68
n	29.	0.59	31.	0.656
v	30.	0.579	29.	0.681
t	31.	0.539	33.	0.608
d	32.	0.531	35.	0.596
el	33.	0.524	38.	0.573
h	34.	0.508	37.	0.58
U	35.	0.5	39.	0.55
em	36.	0.5	28.	0.7
I	37.	0.496	36.	0.585
en	38.	0.476	42.	0.5
zh	39.	0.466	34.	0.6
dh	40.	0.436	41.	0.501
D	41.	0.427	40.	0.502
a	42.	0.423	45.	0.433
E~	43.	0.416	46.	0.416
ae~	44.	0.411	44.	0.47
aa̰	45.	0.391	43.	0.478
o	46.	0.384	32.	0.615

Segment	Rank in Matching Algorithm	Matches / Total (Stability)	Rank in Re-Matching Algorithm	Matches / Total (Stability)
ih	47.	0.318	48.	0.319
q	48.	0.282	47.	0.321
z%	49.	0.265	49.	0.247
k'	50.	0.214	51.	0.214
D~	51.	0.2	50.	0.24
a%	52.	0.133	53.	0.133
d%	53.	0.105	52.	0.157
t'	54.	0.08	55.	0.08
dh%	55.	0.052	54.	0.105
~ai	56.	0.0	56.	0.047

Appendix D
Sample Alignments from Matcher

```
  6 [ m   ai  f   o   k   s   l   I   v   ih  n   m   ih      z   U   r   i   ]
    [         f   o   k   s   l%  I               m       em  z   U   r   i   ]

 15 [ h   au  f   ah  r   I   z   t'  a   p'  i   k'  a   f   r   a   m       k'  ae  n   z%  a   s   s   I   d   i   ]
    [ h   au  f   ah  r       z%  t°      p   i   k   a   f               em  k   ae  n   z       s       I   d   i   ]

 23 [ p   ae  k   ih  s   t   ae  D~  a   z   d   a   v   E   l   a   p   th  i   a   t   ah  m   ih      k   b   ah  m   ]
    [ p   ae  k       s   t   ae~         z   d       v   E   l   a   p   dh          t   ah  m-      i   k&  b   ah  m   ]

 25 [ t   s   q   i   z   i   a   r   d   a   m   uh  l   t   a   p   l   ai  dh  a   n   t   a   d   a   v   ai  d   ]
    [ t   s       i   z           er  D   a   m   uh  l           p   l   ai          n           d   a   v   ai  d   ]

 29 [ w   a   z   jh  o   z   a   f   k   E   D~  a       d   i   a   n   q   ai  r   I       sh  q   I   m   a   g   r   a   n       t°  ]
    [ w   a   z   jh  o   z%      f   k   E           n   d   i   a   n       ai  r       ih  sh      i   m   a   g               er~ q   ]

 55 [ dh  ih      s   m   E   k   s   k   a   n   t   a   k'  i   l   a   I   z   l   I   k   w   a   d   d   ai  n       a   m   ai  q   ]
    [ dh      I   s   m   E   k   s   k   a   n   t°      k'  i   l   a       z   l   I   k   w   a   d       ai      D~  a   m   ai  q   ]

 69 [ n   ou  b   a   D   i   ai  n   ou  h   ae  z   E   v   er  l   I       v   d   ih  n   E   r   a   z   o~  n   a   ]
    [ n   o   b   a   D   i   ai. n   o           z   E   v   er  l       ih  v           n   E   r       z   o   n   a   ]

 78 [ dh  a   p   r   ah  s   a   k       y   u   sh  en  l   ae  k   s   s   a   f   I   sh  a   n   t   E   v   a   d       en  t   s   ]
    [         p   r   ah  s           k-  y   u   sh  en  l   ae  k   s   s       f   I   sh      n       E   v   a   D   a   n       s   ]

103 [ ~ai h   ae  v   n   E   v   er  m   E   q   w   I       dh  a   n   ~E  r   a   b   w   ih      th  a   h   E   r   ih  m   ]
    [ ai          v   n   E   v   er  m   E       w       ih  dh  a   n   E   r       b   w       I   th  a   h   E   r       m   ]

154 [ dh  ih  s   s   E   m   ih  n   ah  r   ~ih z   ae  z   k       r   au  D   a   d   ae  z   n   o   w   a   z   q   ah  r   k   ]
    [ dh  ih  s       E   m       n   ah  r       s               k-  r   au                  z   n   o       a   z       ah  r   k   ]
```