NCC 9-16
IN-32-CR
73464

Syntactic Error Modeling and Scoring Normalization in Speech Recognition: Progress Reports

p-23

Lex Olorenshaw

Speech Systems Incorporated

11/30/91, 1/31/91 & 3/31/91

Cooperative Agreement NCC 9-16
Research Activity No. ET.28

NASA Johnson Space Center
Information Systems Directorate
Information Technology Division

Research Institute for Computing and Information Systems
University of Houston-Clear Lake

INTERIM REPORT

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Lex Olorenshaw of Speech Systems Incorporated. Dr. Glenn Freedman served as RICIS research coordinator.

Funding has been provided by the Information Technology Division, Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was James A. Villarreal, of the Software Technology Branch, Information Technology Division, Information Systems Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.

SSI/UHCL Subcontract – Literacy Tutor Project

First Report:

Specification of Research Methods

1. Introduction

The purpose of the project is to develop our speech recognition system to be able to detect speech which is pronounced incorrectly, given that the text of the spoken speech is known to the recognizer. NASA-JSC will then develop this technology into a "Literacy Tutor" to run on the Mac IIci. The Literacy Tutor will also incorporate other new technologies (such as video input to the Mac) in order to bring innovative concepts to the task of teaching adults to read.

2. Overview of Technical Objectives

The technical objectives of this project are:

1) Develop our system so that when an isolated word is pronounced incorrectly, the recognizer will reject it. The expected word is known to the recognizer before decoding begins.

Example-1a:

SYSTEM PROMPTS: say this word - "cat".
SPEAKER SAYS: [kæt] ("cat").
SYSTEM RESPONDS: pronounced correctly.

Example-1b:

SYSTEM PROMPTS: say this word - "cat".
SPEAKER SAYS: [kout] ("coat").
SYSTEM RESPONDS: pronounced incorrectly.

2) Investigate how our system can provide information/feedback as to which part/phoneme(s) of an incorrectly pronounced word has been pronounced poorly.

Example-2:

SYSTEM PROMPTS: say this word - "cat".
SPEAKER SAYS: [kout] ("coat").
SYSTEM RESPONDS: "pronounced incorrectly. [æ] was poorly pronounced (as [ou])."

We feel that if our system can reliably accomplish these two tasks, it will provide a *very* valuable tool to the Literacy Tutor. Further utility of the speech recognizer would come as result of accomplishing the following objectives:

3) Develop our system so that when a multi-word utterance is spoken incorrectly into the recognizer, the system can reject it as being pronounced incorrectly.

Example-3:

SYSTEM PROMPTS: say this sentence - "the cat says meow".
SPEAKER SAYS: [ðə kout sɛz miaw] ("the coat says meow")
SYSTEM RESPONDS: sentence pronounced incorrectly.

4) Investigate how our system can provide information/feedback as to which word of an incorrectly pronounced utterance has been poorly pronounced.

Example-4:

SYSTEM PROMPTS: say this sentence - "the cat says meow".
SPEAKER SAYS: [ðə kout sɛz miaw] ("the coat says meow")
SYSTEM RESPONDS: sentence pronounced incorrectly. "cat" was poorly pronounced.

5) As an extension of objectives 2) and 4), investigate how our system can provide information/feedback as to which phones within incorrectly pronounced words (within an incorrectly pronounced utt) have been poorly pronounced.

Example-5:

SYSTEM PROMPTS: say this sentence - "the cat says meow".
SPEAKER SAYS: [ðə kout sɛz miaw] ("the coat says meow")
SYSTEM RESPONDS: sentence pronounced incorrectly. The word "cat" was poorly pronounced. (Within the word "cat") [æ] was poorly pronounced (as [ou]).

3. Background

The original proposal outlines two methods for proceeding with this work. The first method is "Syntactic Error Modelling"; the second is "Score Normalization". Depending on the preliminary results of these efforts, we may also investigate third method called "Phoneme Error Modelling". Each of these methods is described briefly in the sections below.

3.1 Syntactic Error Modelling

The original purpose of this project was to provide a quick and easy way for our system to accomplish objective 1. It was thought that if the types of reading errors that are made can be modelled as word errors (e.g. "cat" pronounced as "coat"), then the syntax can provide a way for errors to be detected by the recognizer. The success of this error-modelling technique would depend on: 1) how many of the errors made can be modelled as word errors, and 2) how well our recognizer can distinguish the word errors.

3.2 Score Normalization

The original proposal contained an explanation of the "Score Normalization" project which would be done to get the recognizer to produce decoding scores which would approximate "goodness of pronunciation" judgements of humans. In other words, scores output by the decoder could produce better confidence thresholds to correctly reject mispronounced words. For example, if the user/student says the word "cat" as [kæt], you would like the word and/or utt score to be such that it would always be above some rejection threshold.

On the other hand, if the user/student says the word "cat" as "coat", you would like the word and/or utt score to be such that it would always be below some rejection threshold. In the past, this has not always been the case. "Score Normalization" was conceived as being a way to have the scores be reliable for accurate acceptance/rejection.

3.3 Phoneme Error Modelling

This method was not outlined in the original proposal, but we believe that it may prove useful in our efforts to provide feedback as to which sounds within a word have been poorly pronounced. The above-mentioned methods inherently do not have any way of providing information at the sub-word level. Therefore, if we are to provide sub-word level feedback regarding mispronunciations, then we will need a method to do so. In general, this method calls for experimenting with the phoneme representation of words in the phonetic dictionary used by the recognizer. By specifying potential phoneme level errors in the entries of the phonetic dictionary, the speech recognition system will have an opportunity to select a sequence of phonemes which more accurately represents the mispronounced word.

4. Methodology

4.1 Syntactic Error Modelling

We have recently had some experience with this approach in research performed on “keyword spotting”. In the keyword task, the speech recognizer tries to isolate only those words which are thought to have some key meaning. The developer provides a list of keywords to be recognized, as well as a list of potential non-keywords. When a sentence of speech is input into the system, the recognizer attempts to filter keywords from non-keywords, and then display the keywords which were recognized.

This is similar to syntactic error modelling in that for each word a student will read aloud into the recognizer, we would like to have a listing of words which are often spoken as mispronunciations of the prompted word. This list of words we call *miscue* words. The recognition system will utilize this information as it tries to determine whether or not the student read the word(s) correctly. By knowing the potential errors that the student will make, the recognizer can consider the potential sequences of phonemes which may have been spoken, even if the word has been mispronounced as another word.

4.1.1 Activities

An outline of the tasks for this method is presented below. Note that we will need to create a comparison case to measure the effectiveness of using real world word errors. This will be done by randomly choosing a set of words to act as the *miscue* words.

For objective 1:

1) Define/design a test case for isolated word recognition
2) Investigate what the possible word errors are for the test case.
3) Collect test data
4) Create syntaxes using the potential word errors as *miscue* words.
5) Test the performance of the system for correct hits, correct rejections, incorrect rejections, etc.
6) Create syntaxes using randomly chosen words as *miscue* words.

7) Test the performance of the system for correct hits, correct rejections, incorrect rejections, etc.; compare results with above tests which used real world word errors.

For objective 2:

8) Examine the results to see how well the system performed in choosing a correct transcription from all possible *miscue* words to match an incorrectly pronounced word.

For objective 3:

9) Define/design a test case for utterance recognition
10) Investigate what the possible word errors are for the test case.
11) Collect test data
12) Create syntaxes using the potential word errors as *miscue* words.
13) Test the performance of the system for correct hits, correct rejections, incorrect rejections, etc.
14) Create syntaxes using randomly chosen words as *miscue* words.
15) Test the performance of the system for correct hits, correct rejections, incorrect rejections, etc.

For objective 4:

16) Examine the results to see how well the system performed in choosing any *miscue* word to align with an incorrectly pronounced word.

For objective 5:

17) Examine the results to see how well the system performed in choosing a correct transcription from the *miscue* words to match an incorrectly pronounced word.

4.1.2 Issues

We must be cautious as to how well any results we obtain will be representative of the entire range of reading/pronunciation errors. It is unclear at this point how wide the range of errors might be throughout various reading levels/capabilities. Furthermore, it is unknown whether or not various readers within a given level will produce similar reading/pronunciation errors.

In addition, given the current state of how well the decoder is able to make close-phoneme distinctions, it seems inevitable that this approach will have its limitations in terms of implementation in a live testing/tutoring system.

However, this phase of the project seems essential in order to gather some information on the types of pronunciation errors that the recognizer will need to distinguish, and to get a baseline of how the recognizer can perform given the current technology.

4.2 Score Normalization

During the process of decoding the input speech, the Phonetic Decoder produces scores for the words it is considering. The word sequence with the highest total score is chosen as the output word sequence. The score of a word is a measure of how well some portion of the input speech matched with the Decoder's internal model of that word. Thus it seems reasonable that this score (in some form) can be used to evaluate the quality of the pronunciation of a word.

However, these scores are not normalized. That is, the distribution of the scores will be different for different words. The most obvious difference among scores for different words comes from the word length. Longer words will have more terms in their scores, on the average, than shorter words. This makes the scores of short and long words incomparable. Also, some phonemes are better recognized than others, which makes the scores for words with well recognized phonemes have a higher potential than the scores for words with poorly recognized phonemes.

The Decoder avoids most of these problems by only comparing scores corresponding to the same range of the speech input. This cannot be done in a pronunciation evaluation application, because we have to be able to compare different instances of the same word (and different words), that is, different speech input, on some comparable scale.

Score normalization seeks a way to normalize the scores for different instances of different words, so that they are comparable in an absolute sense, rather than in the relative sense that they are now. Better scores should then correspond to better matches with internal word models, which should in turn correspond to better word pronunciations.

4.2.1 Activities

We propose a six step process for preparing a scoring normalization technique:

1) Measure the nature of the word score distributions.
2) Analyze the phenomena creating the differences among these distributions.
3) Prepare a normalizing method addressing the known differences.
4) Implement the normalizing method.
5) Test the normalizing method.
6) Depending on the results from these preliminary investigations, consider how score normalization could be implemented into the runtime speech recognizer and the literacy tutor application.

For step one, we will improve our analysis tools for word scores to plot the distributions of word scores. From this we can measure the degree of non-normalization present in the raw words scores, and evaluate the improvement resulting from any normalization method to be implemented.

Step two considers the factors that may be influencing the word score distributions that make them non-normalized. This analysis is to develop an intuition into what will be important in a method to normalize the scores.

The third step is one of coming up with a normalization method. Step four implements this method, which is tested in step five. In step six we conclude how useful the normalization method is for the literacy tutor application.

4.2.2 Issues

It is perhaps worthwhile to mention what the ideal word score distribution would look like.

First of all, all scores for a word matched with a region of input where that word was actually spoken should be higher than all scores for that word matched with a region where that word was not spoken. Thus we have two separate sub-distributions of word scores for a word, one where the word was spoken and one where the word was not. These sub-distributions should be cleanly separated by a word acceptance (or rejection) threshold, so that the word score can be used to see if the word was correctly matched.

Within the sub-distributions, the scores should correspond to the quality of the pronunciation of the word for the correct matches, and some pronunciation similarity for the incorrect matches.

4.3 Phoneme Level Error Modelling

The Phonetic Decoder software requires two main knowledge sources: the phonetic dictionary and the syntax (or grammar). By considering the types of phonetic errors that occur ("miscue analysis") we should be able to provide a model of these errors to the recognizer via the phonetic dictionary. This could be done for each word to be used in the reading application. However, it may also be that there is a way to more globally indicate the range of potential phonetic errors to the recognizer without having to consider the specific errors for each word to be recognized. This could be done by considering the phonotactic rules of English which constrain the occurrences of phonemes in context. If a *meta-word* can be designed which adequately models these constraints, then it may provide a way of modelling phonetic errors which can be used for all words under consideration. For example, at a rather course level a meta-word to represent many one-syllable words could be constructed as [(C)(G)V(G)(C)], where C=consonant, G=glide and V=vowel. Parentheses indicate optional phonetic entities. A more complex meta-word to model one syllable words of English could be [{(F)({N|S})(G)|(H)}V(G(G))({N|S|NS})(F({S|F})] where F=fricative, N=nasal, S=stop, G=glide, H="h" and V=vowel. Curly braces indicate either/or options, separated by the vertical bar "|".

4.3.1 Activities

1) Examine the phonetic errors made in reading tasks (i.e. miscue analysis).
2) Design a test.
3) Create phonetic error models for specific words.
4) Create *meta-words* to model phonetic errors.
5) Test utility of specific word phonetic error models vs. *meta-word* phonetic models.
6) Depending on results of preliminary tests, consider how phoneme modelling can be implemented into the runtime recognition system and literacy tutor application.

4.3.2 Issues

The above examples of meta-words assume that the student would not speak a multi-syllable word. Depending on the results of our investigation of the types of errors that readers will make, this may or may not be a valid assumption. It does seem that modelling only one-syllable words could turn out to be a reasonable case for certain aspects of the Literacy Tutor application. We may also need to consider making more complex meta-words to model multi-syllable words, etc. We should also consider how strictly the meta-word should follow the phonotactic rules of English.

5. Summary

In order to develop our speech recognizer to be able to detect speech which is pronounced incorrectly, we will perform research in three areas: 1) syntactic error modelling; 2) score normalization; and 3) phoneme error modelling. Our investigations into the types of errors that a reader makes (i.e. miscue analysis) will provide the basis for creating tests which will approximate the use of the system in the real world. Depending on the success of our preliminary investigations, we will consider how each of these methods can be integrated into our runtime speech recognition system and the literacy tutor application.

SSI/UHCL Subcontract – Literacy Tutor Project

Second Progress Report

This report summarizes the work performed by Speech Systems Incorporated from December 1, 1990 to January 31, 1991.

During this time period we have begun our efforts on the first two objectives outlined in the Specification of Research Methods. These objectives are:

1) Develop our system so that when an isolated word is pronounced incorrectly, the recognizer will reject it. The expected word is known to the recognizer before decoding begins.

Example-1a:

SYSTEM PROMPTS: say this word - "cat".
SPEAKER SAYS: [kæt] ("cat").
SYSTEM RESPONDS: pronounced correctly.

Example-1b:

SYSTEM PROMPTS: say this word - "cat".
SPEAKER SAYS: [kout] ("coat").
SYSTEM RESPONDS: pronounced incorrectly.

2) Investigate how our system can provide information/feedback as to which part/phoneme(s) of an incorrectly pronounced word has been pronounced poorly.

Example-2:

SYSTEM PROMPTS: say this word - "cat".
SPEAKER SAYS: [kout] ("coat").
SYSTEM RESPONDS: "pronounced incorrectly. [æ] was poorly pronounced (as [ou])."

Syntactic Error Modelling

Work on the following tasks is described in more detail below:

For objective 1:

1) Define/design a test case for isolated word recognition.
2) Investigate what the possible word errors are for the test case.
3) Collect test data.
4) Create syntaxes using the potential word errors as *miscue* words.
5) Test the performance of the system for correct hits, correct rejections, incorrect rejections, etc.
6) Create syntaxes using randomly chosen words as *miscue* words.
7) Test the performance of the system for correct hits, correct rejections, incorrect rejections, etc.; compare results with above tests which used real world word errors.

For objective 2:

8) Examine the results to see how well the system performed in choosing a correct transcription from all possible *miscue* words to match an incorrectly pronounced word.

Preliminary Testing

A baseline test case for syntactic error modelling has been completed validating that this method can provide acceptable results. This test case was put together to give a quick idea of the feasibility of this method. First, a test case for isolated word recognition was chosen. We chose the first lesson from the Sight Words 2 Workbook, the second booklet in a series from the *TV Tutor®*. The reading lessons from this series are all isolated word reading tests of "sight" words. These are words that occur frequently in written English and that efficient readers recognize easily. Each lesson contains six words for students to read aloud, spell, etc. There are ten lessons for a total of 60 words in the workbook.

In this testing scenario, when each of the six words from Lesson 1 is being tested for accuracy, the remaining five words from Lesson 1 serve as the miscue words. The five miscue words and the remaining 54 words from the workbook serve as the non-test words which are listed in a syntax for recognition.

The test words from Lesson 1 are: *round, must, under, any, pretty* and *open*. We want to know not only the accuracy rates for each of these words, but also the correct rejection or false alarm rates. (In this scenario, each incorrect rejection is a false alarm.) Correct rejection rates tell us how often a non-test word is successfully recognized as any non-test word. False alarm rates (100% – CorRej%) indicate how often the miscue word is incorrectly recognized as the test word.

Investigation of Word Errors

As yet, we have not been able to locate any published source that provides the results of reading tests by categorizing various words and the miscue words associated with them (i.e expected response vs. observed response). Initially, we had checked several "dictionaries" which provide information on "confusing" or "mispronounced" words. Most of these present only a descriptive account of problem words, usually pairs of words, which are thought to be confused by speakers of the English language. They give little indication of which words are problems for *readers*.

Much of the research in miscue analysis performed over the past three decades is in the form of unpublished dissertations. Therefore, it is not easily accessible. However, we have contacted several university researchers across the nation to find out if current work is being done that may provide more information. Dr. Ken Goodman at the University of Arizona in Tuscon has recently been involved in a study to examine word level miscues. He has promised to send us a listing of words and the miscue words from a recent study of approximately 30 readers from grades 2, 4 and 6. We anticipate that this listing will give us the opportunity to see how our system performs with actual miscue words, rather than our current test miscue words described above.

Data Collection

To generate test data we collected 10 repetitions of each of the six words in Lesson 1. This test data set was collected by each of three male adult speakers, giving us a total of 180 test tokens. These tokens are used in 6 different test situations to examine the performance of each as the "expected response" word. The recognition speaker model used to collect the data is the R3.4 Generic Male model, 3013.

Test Syntaxes

Six test syntaxes have been created to test how each of the six words performed with the recognizer. As mentioned above, when one of the six words is the test word, the other 59 words from the workbook serve as the potential observed responses in the syntax. Below is the syntax to use when *round* is the test word.

```
S -> { TESTWORD | POTENTIAL_OBSERVED }

TESTWORD == round

POTENTIAL_OBSERVED ==
must
under
any
pretty
open
today
been
goes
night
walk
soon
boy
there
call
may
find
look
these
give
which
read
school
want
why
keep
milk
does
bird
ready
take
back
use
book
four
those
don't
birthday
laugh
friend
please
small
start
our
```

```
other
much
could
circle
every
thank
where
because
ate
always
know
hurry
sure
done
answer
own
```

The test syntaxes for each of the other words is made by replacing *round* with the new test word. At the same time, *round* is placed into the POTENTIAL_OBSERVED category, and the new test word is removed from the POTENTIAL_OBSERVED list.

Test Results

Tests for each of the six words were run initially on one speaker with the following results:

Test Word	Accuracy	CorRejectPO
any	40%	96%
must	70%	100%
open	100%	100%
pretty	50%	100%
round	40%	100%
under	80%	100%

Table 1 - Initial test with speaker LSO.

"Accuracy" is the recognition accuracy of the test word. "CorRejectPO" is the correct rejection rate of the five other words serving as miscue words. After observing these results, some minor changes were made to the syntaxes to remove POTENTIAL_OBSERVED words which were too often confused as test words. These words were:

`any` (except in the syntax for the test word *any*)

```
ate
today
take
every
night
bird
ready
don't
own
friend
keep
may
find
```

```
those
give
read
```

In addition, one change was made to the phonetic spelling of *round* in the recognition dictionary (by adding the [æ] vowel as an option to [a] in the [aω] diphthong). These minor changes improved accuracy for the test words while keeping the rejection rates to an acceptable level when tested with speaker LSO. Several more words were removed from the syntaxes to improve accuracy for two more speakers. These words were:

```
always
answer
please
```

The results of all three speakers using the final revised version of the syntaxes are displayed in Tables 2, 3 and 4. In addition to tests of correct rejection of words in the POTENTIAL_OBSERVED list, we also tested the correct rejection of words not in the syntax. These were six words taken from Lesson 10 of the Sight Words 1 Workbook: *again, would, very, or, many* and *only*. Each speaker donated five repetitions of each of the six words for this test. The rates of correct rejection for these words are in the "CorRejNonPO" column.

Test Word	Accuracy	CorRejectPO	CorRejNonPO
any	90%	94%	83%
must	100%	100%	100%
open	100%	100%	100%
pretty	90%	94%	87%
round	90%	100%	100%
under	100%	98%	97%

Table 2 - Revised test with speaker LSO.

Test Word	Accuracy	CorRejectPO	CorRejNonPO
any	90%	100%	90%
must	100%	100%	100%
open	90%	100%	90%
pretty	80%	100%	90%
round	100%	98%	93%
under	100%	98%	100%

Table 3 - Revised test with speaker BMD.

Test Word	Accuracy	CorRejectPO	CorRejNonPO
any	100%	100%	80%
must	100%	100%	100%
open	100%	100%	100%
pretty	100%	92%	93%
round	80%	100%	93%
under	80%	100%	93%

Table 4 - Revised test with speaker DJT.

These results indicate that the recognition system has the ability to accurately recognize a word when pronounced correctly. It is also able to fairly reliably reject a set of miscue words when the expected response is the test word. Whether the set of miscue words is representative of the types of actual miscues made by readers will be the focus of upcoming tests.

Phoneme Error Modelling

We have also performed some preliminary experiments with phoneme error modelling. The first step was to create a *meta-word* that would contain phonetic pronunciations for any one-syllable word to be spoken. This first attempt produced a phonetic transcription which our dictionary compiler could not generate since the resulting dictionary "graph" was too large. To reduce the size of the dictionary graph, we divided the entire one-syllable meta-word into several separate meta-words. These several meta-words, when used as a set, would cover the entire range of phonetic transcriptions as the original meta-word design. All but one of the smaller meta-words was able to be compiled successfully.

We have performed some informal preliminary tests of the meta-word concept with these newly generated dictionary entries. It appears that in their current form the meta-words provide too many phonetic transcriptions for the recognizer to successfully distinguish a test word from a miscue meta-word. We will be continuing our investigation of this concept by trying to reduce the number of phonetic spellings contained in the meta-words. This reduction will be done by considering the phonotactic constraints currently inherent in the English language.

Score Normalization

No progress has been made on this project.

Demo System

In looking towards the opportunities we may have to demonstrate this technology, we have created prototype application software to run live speech recognition in this isolated word testing mode. We will be expanding the capabilities of this software as we learn more about how this technology will be implemented in the Intelligent Reading Training System under development by the Software Technology Branch at NASA Johnson Space Center.

SSI/UHCL Subcontract – Literacy Tutor Project

Third Progress Report

This report summarizes the work performed by Speech Systems Incorporated from February 1, 1990 to March 31, 1991.

During this time period we have continued our efforts on the first two objectives outlined in the Specification of Research Methods. These objectives are:

1) Develop our system so that when an isolated word is pronounced incorrectly, the recognizer will reject it. The expected word is known to the recognizer before decoding begins.

   Example-1a:

   SYSTEM PROMPTS: say this word - "cat".
   SPEAKER SAYS: [kaet] ("cat").
   SYSTEM RESPONDS: pronounced correctly.

   Example-1b:

   SYSTEM PROMPTS: say this word - "cat".
   SPEAKER SAYS: [kout] ("coat").
   SYSTEM RESPONDS: pronounced incorrectly.

2) Investigate how our system can provide information/feedback as to which part/phoneme(s) of an incorrectly pronounced word has been pronounced poorly.

   Example-2:

   SYSTEM PROMPTS: say this word - "cat".
   SPEAKER SAYS: [kout] ("coat").
   SYSTEM RESPONDS: "pronounced incorrectly. [ae] was poorly pronounced (as [ou])."

Syntactic Error Modelling

Investigation of Word Errors

Work on this aspect of the project is on hold until we can define a set of test words which represents a set of actual *miscue* words. Our initial contact with Dr. Ken Goodman at the University of Arizona seemed quite promising. He had indicated that he would send us a listing of words and the miscue words from a recent study of approximately 30 readers from grades 2, 4 and 6. However, we have since been unable to discuss this matter further with him, and it appears that he will not be sending us the word lists any time soon.

Therefore, we are resuming our search for material contained in dissertations, and hope to find suitable information in them soon.

Phoneme Error Modelling

We have continued our investigations into the creation of a *meta* word representation which could be useful in determining the phonetic level errors which are made in pronunciation. We designed a scheme of phonotactic constraints which would be present in all one-syllable words of the English language. The basic scheme can be described as follows: the presence of at least one vowel, which is optionally preceded and/or followed by one or a sequence of (phonotactically legal) consonants. This can be displayed by the following set of expansion rules:

1-Syllable = ({ G_i | C_i | K_i })
{ V_o (K_f)
| V_y G_y (K_y)
| V_{wy} { G_w (K_w) | G_y (K_y) }
}

where:

G_i = word initial glides
C_i = word initial single consonants
K_i = word initial consonant clusters (i.e. any legal combination of glides and consonants)
V_{wy} = "low back A" vowel (which precedes /y/ and /w/ in the diphthongs of "buy" and "cow" respectively)
V_y = "open O" vowel (which precedes /y/ in the diphthong of "boy")
V_o = all other vowels (except "open O" and "low back A")
G_y = the /y/ glide (which is word-final in "boy" and "buy")
G_w = the /w/ glide (which is word-final in "cow")
K_f = final single consonants except /y/ and /w/, and consonant clusters which do <u>not</u> begin with /y/ or /w/
K_y = the consonant clusters which can follow the /y/ glide
K_w = the consonant clusters which can follow the /w/ glide
() = contents are optional
{} = contents are and "either/or" choice
| = choice separator

It was desirable to make some distinction for the /y/ and /w/ glides so that they would combine appropriately with the "open O" vowel and the "low back A" vowel. Note that in the designing of this scheme, we only considered the phonotactic constraints that occur in the Western American dialect of English. This is due to the fact that our phonetic transcription representation does hold to some particulars in symbology which are consistent with this dialect. Some phonotactic combinations not allowed here might be considered appropriate for a representation of other (American) English dialects.

Once this scheme was designed, we attempted to implement it into the ASCII graph notation of our phonetic dictionary. This required the creation of a structure for a single dictionary entry which contained multiple phonetic representations. However, the tool to compile the ASCII representation into a binary file was unable to handle the size of the resulting dictionary graph. Therefore, we decided that we could implement the same phonotactic rule scheme outside of the dictionary by using the syntax phrase rule technique. Each phonetic element of the phonetic dictionary transcription set would need to be represented in the phonetic dictionary as a unique word. These "words" were then used to construct the phonotactically correct one-syllable meta-word via syntax phrase rules. The resulting phrase rule structure is as follows:

```
S -> 1SYLLABLE_

1SYLLABLE_ -> ( { Gi | Ci | Ki } )
                         { Vo (Kf)
                         | Vy Gy(Ky)
                         | Vwy { Gw(Kw) | Gy(Ky) } }


Vo ==  Anoth Aacute Aquotes aschwa Enoth Eacute
        eschwa Iacute Inoth Onoth Oacute Unoth Uacute

Vy == Ograve

Vwy == Agrave

Ci == dh_ f_ h_ s_ sh_ th_ v_ z_ zh_ m_ n_
      cx_ kx_ px_ tx_ bx_ dx_ gx_ jx_ q_

Cf == dh_ f_ s_ sh_ th_ v_ z_ zh_ m_ n_ ng_
      cx_ kx_ px_ tx_ bx_ dx_ gx_ jx_

Gi == l_ r_ w_ y_

Gw == w_

Gy == y_


Ki -> {
{(s_)bx_|f_|px_}{l_|r_|y_}
|{dx_|tx_}{r_|w_|y_}
|{(s_)gx_|kx_}{l_|r_|w_|y_}
|h_{w_|y_}
|th_{r_|w_}
|s_ dx_ {r_|y_}
|{l_|(s_)m_|n_|s_|v_}y_
|sh_ r_
|s_{f_|l_|m_|n_|w_}
}


Ky -> {
jx_(dx_)
|dx_(z_)
|s_(tx_(s_))
|{bx_|m_|v_}({dx_|z_})
|n_{{tx_|th_}(s_)|({dx_|z_})}
|dh_({dx_|z_})
|l_({dx_(z_)|z_})
|{px_|kx_}({s_|tx_})
|tx_(s_)
|r_ z_
}
```

```
Kw ->
{
jx_(dx_)
|dx_(z_)
|s_(tx_(s_))
|cx_(tx_)
|dh_({dx_|z_})
|l_({dx_|z_})
|n_(tx_(s_(tx_))|({dx_|z_}))
|th_({s_|tx_})
|tx_(s_)
}

Kf -> { K_LRf | K_STOPf | K_FRICf }

K_LRf -> { LONLY_ | RONLY_ | (LRBOTH_) }

LONLY_ -> l_({tx_ s_ tx_|f_({s_|th_(s_)})})

RONLY_ -> r_({n_ tx_ (s_)|l_({dx_(z_)|z_})|dh_{dx_|z_}|gx_({dx_|z_})})

LRBOTH_ ->
{
{l_|r_}{{jx_(dx_)|dx_(z_)|n_({dx_|z_})|s_(tx_(s_))}
        |{bx_|m_|v_}({dx_|z_})|cx_(tx_)|{px_|kx_|th_}({s_|tx_})
        |sh_(tx_)|tx_(s_)|z_}
}

K_FRICf ->
{ dh_ ({dx_|z_})
| f_ ({s_|tx_(s_)|th_({s_|tx_})})
| s_ ({{px_|kx_}({s_|tx_})|tx_(s_)})
| sh_ (tx_)
| th_ ({s_|tx_})
| v_ ({dx_|z_})
| z_ (dx_)
| zh_ (dx_) }

K_STOPf ->
{ m_ ({dx_|{px_|f_}({s_|tx_})|z_})
| n_ ({dx_(z_)|th_({s_|tx_})|s_(tx_)|sh_(tx_)|zh_(dx_)|tx_(s_)|z_})
| ng_ ({dx_|kx_({s_|tx_})|z_})
| cx_ (tx_)
| kx_ ({s_({tx_|th_(s_)})|tx_(s_)})
| px_ ({s_(tx_)|tx_(s_)})
| tx_ ({s_(tx_)|th_({s_|tx_})})
| bx_ ({dx_|z_})
| dx_ (z_)
| gx_ ({dx_|z_})
| jx_ (dx_) }
```

The phoneme "words" as named above correspond to the SSI phonetic transcription representation as follows:

```
Anoth = /A/
Aacute = /A'/
Aquotes = /A"/
aschwa = /a/
```

```
Enoth = /E/
Eacute = /E'/
eschwa = /e/
Iacute = /I'/
Inoth = /I/
Onoth = /O/
Oacute = /O'/
Unoth = /U/
Uacute = /U'/
Ograve = /O`/
Agrave = /A`/
dh_ = /d!/
f_ = /f/
h_ = /h/
s_ = /s/
sh_ = /s!/
th_ = /t!/
v_ = /v/
z_ = /z/
zh_ = /z!/
m_ = /m/
n_ = /n/
ng_ = /n;/
cx_ = /c/ (released)
kx_ = /k/ (released)
px_ = /p/ (released)
tx_ = /t/ (released)
bx_ = /b/ (released)
dx_ = /d/ (released)
gx_ = /g/ (released)
jx_ = /j/ (released)
q_ = /q/
l_ = /l/
r_ = /r/
w_ = /w/
y_ = /y/
```

The above phrase rule syntax is able to accept/generate sequences of phonemes which are phonotactically correct for a one-syllable word in English. Below are a few examples. If the phonemic representation corresponds to an actual English word, then the orthography (i.e. spelling) of that word is shown. Otherwise, a hypothetical orthography is shown.

joy:	jx_ Ograve y_
guy:	gx_ Agrave y_
myah:	m_ y_ Aacute
prove:	px_ r_ Uacute v_
gyoip:	gx_ y_ Ograve y_ px_
vyoy:	v_ y_ Ograve y_
froit:	f_ r_ Ograve y_ tx_
thrigh:	th_ r_ Agrave y_
pyow:	px_ y_ Agrave w_
choinths:	cx_ Ograve y_ n_ th_ s_

Although this generates phoneme sequences for real English words, it also generates sequences which are not "real" words. However, these words are considered "pronounceable" due to the phonotactic constraints which have been incorporated into the rules that generate them. We are now in the process of reviewing the types of phoneme sequences it generates to ensure that there are no phonotactic sequences which are "illegal" with respect to the English language.

We have also performed some preliminary recognition tests (on a very limited test set) to see how well the recognition system using the meta-word syntax can produce "correct" phoneme sequences. The "correct" phoneme sequences for each word spoken were taken from the phonemic transcription of that word contained in our phonetic dictionary. For one-syllable words, we have seen about a 60% accuracy (phoneme-by-phoneme) in matching phoneme sequences. This is very encouraging and leads us to believe that we will be successful in providing useful feedback regarding the sub-word level content of mispronounced words.

Score Normalization

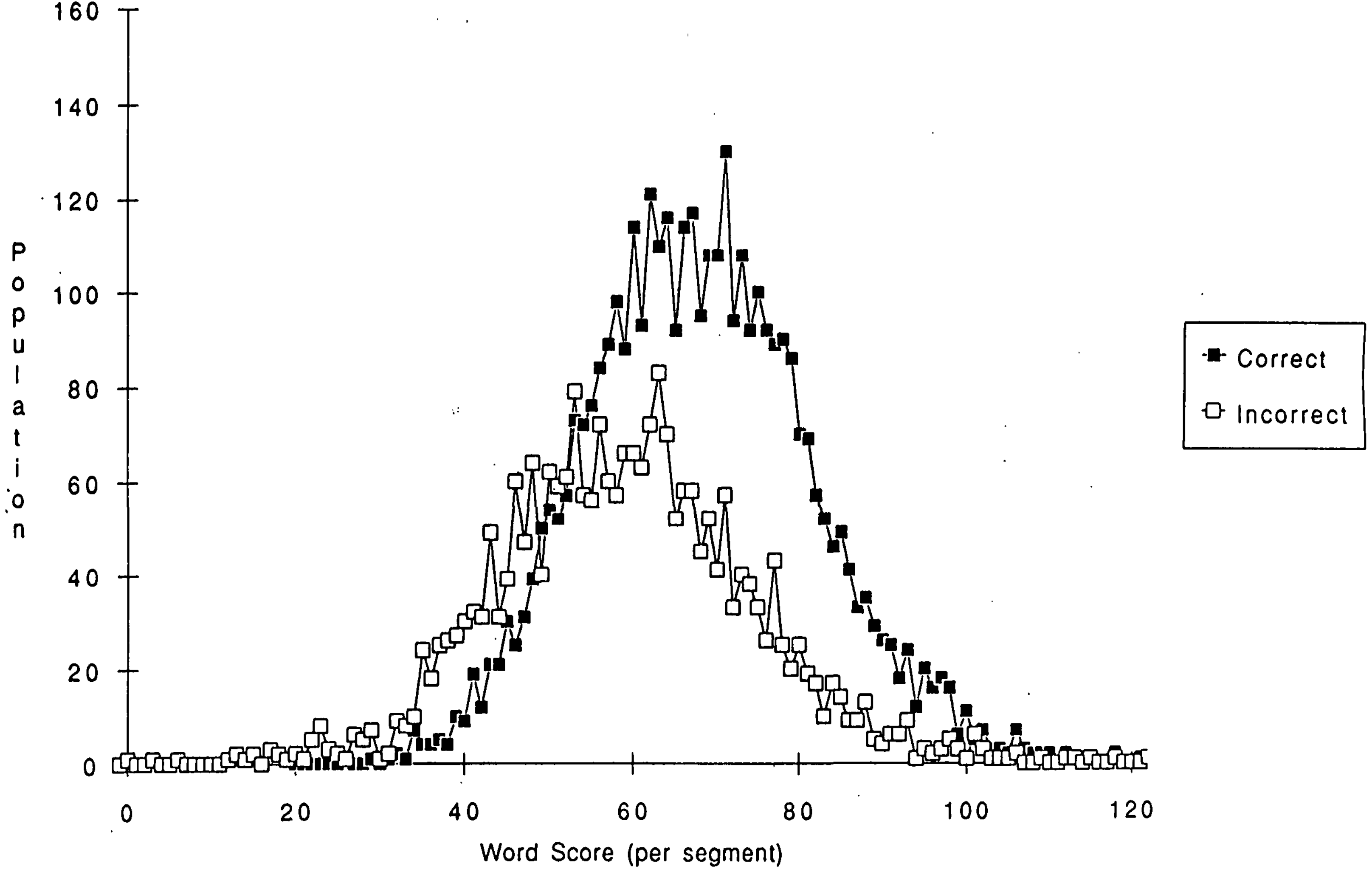
The first phase of this effort has begun. We are working on a preliminary evaluation of word score normalization before and after various tunings. These tuning techniques include phonetic codebook tuning, adjustment of the language weight in the decoder, and a combination of the two.

We have also updated a software tool which will allow for easier evaluation of the resulting research data. This tool will allow us to display a graph of word scores in order to see how the scores are distributed. An example of the resulting graph is shown on the next page. The sequence of black squares traces the number of correct word items by word score. The white squares trace the incorrect words by word score. The peak of the black curve is slightly to the right of the white curve's peak. Since the peaks (and curves) are not very well distinguished along the word score axis, this indicates that the current word score method is not very well suited to be used as a distinguisher between correct and incorrect words. We will be performing several tests to examine more closely the behavior of the word scores. Then we will be attempting to discover a method which will increase the distinction of word scores for correct versus incorrect decoding.

Demo System

We supplied a preliminary version of demonstration software to the staff at NASA Johnson Space Center who are working on the Literacy Tutor project. We consulted with them on how they could integrate this speech recognition application into a demonstration which would utilize the Macintosh to control the active recognition syntax. We then created an enhanced version of the speech application to also handle receiving information from a serial line (which would be connected to the Macintosh). The executable and source code for this sample program was shipped for them to prepare for their March 12th demonstration.

Word Score Distribution: Prior to any Tuning (3015)


Population
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120
Word Score (per segment)
Correct
Incorrect