Syntactic Error Modeling and Scoring Normalization in Speech Recognition
Final Report:
Error Modeling and Scoring Normalization in the Speech Recognition Task for Adult Literacy Training

Lex Olorenshaw
David Trawick

Speech Systems Incorporated

11/31/91

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

TECHNICAL 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.

Syntactic Error Modeling and Scoring Normalization in Speech Recognition Final Report: Error Modeling and Scoring Normalization in the Speech Recognition Task for Adult Literacy Training

Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Lex Olorenshaw and David Trawick 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.

Error Modelling and Scoring Normalization in the Speech Recognition Task for Adult Literacy Training

by

Lex Olorenshaw
David Trawick

Speech Systems Incorporated
18356 Oxnard Street
Tarzana, California 91356
(818) 881-0885

SSI/UHCL Subcontract – Literacy Tutor Project

Final Report

1. Introduction

The purpose of the project was 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. This is to be used by the staff at NASA-JSC to incorporate this technology into a "Literacy Tutor" multi-media system. The Literacy Tutor will also utilize other new technologies (such as video input) in order to bring innovative methods to the task of teaching adults to read.

2. Overview of Technical Objectives

The technical objectives of this project were as follows:

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 (AUDIO): 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 felt that if our system could reliably accomplish these two tasks, it would 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]).

As a result of the work performed on this project, we were able to achieve success in all but the fifth objective.

3. Methodology

The proposal had proposed two methods for performing this work. The first method was titled "Syntactic Error Modelling"; the second was "Score Normalization". After the contract began, we also began to investigate a third method to achieve our objectives, "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 depended 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.

We 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 can 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 makes, the recognizer can consider the potential sequences of phonemes which may have been spoken, even if the word has been mispronounced as another word.

3.1.1 Activities

An outline of the tasks for this method is presented below. It was necessary to create a comparison case to measure the effectiveness of using real world word errors. This was done initially 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.

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. "Score Normalization" was conceived as being a way to have the scores be reliable for accurate acceptance/rejection.

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 seemed reasonable that this score (in some form) could be used to evaluate the quality of the pronunciation of a word.

However, these scores previously were not normalized. That is, the distribution of the scores was different for different words. The most obvious difference among scores for different words stemmed from the word length. Longer words have more terms in their scores, on the average, than shorter words. This made 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 could not 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 sought a way to normalize the scores for different instances of different words, so that they would be comparable in an absolute sense, rather than in the relative sense that they were previously. Better scores would then correspond to better matches with internal word models, which would in turn correspond to better word pronunciations.

3.2.1 Activities

We proposed 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 would improve our analysis tools for word scores to plot the distributions of word scores. From this we could 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 was to consider the factors that may be influencing the word score distributions that make them non-normalized. This analysis was to develop an intuition into what would be important in a method to normalize the scores.

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

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.

3.3 Phoneme Error Modelling

This method was not outlined in the original proposal, but we believed that it would prove useful in our efforts to provide feedback as to which sounds within a word are poorly pronounced. The above-mentioned methods inherently do not have any way of providing information at the sub-word level. Therefore, if we were to provide sub-word level feedback regarding mispronunciations, then we would need a method to do so. In general, this method called 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 would have an opportunity to select a sequence of phonemes which more accurately represents the mispronounced word.

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 planned to be able to provide a model of these errors to the recognizer via the phonetic dictionary. Theoretically, this would be done for each word to be used in the reading application. However, we anticipated that there would be 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. A *meta-word* could be designed which adequately models these constraints, and would thus 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 "|".

3.3.1 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. Results

In order to develop our speech recognizer to be able to detect speech which is pronounced incorrectly, we performed research in three areas: 1) syntactic error modelling; 2) score normalization; and 3) phoneme error modelling. Due to the success of these results, the acceptance/rejection techniques, along with score normalization, have been incorporated into the runtime speech recognition software. A sample demonstration application shows how these mechanisms can be utilized in an oral reading activity using speech recognition. The *metaword* syntax (and supporting phonetic dictionary) will also provide a way to get phoneme-level information about the spoken speech. However, it appears the accuracy of this output will only be functional for one-syllable words.

4.1 Syntactic Error Modelling Results

4.1.1 Preliminary Testing

A baseline test case for syntactic error modelling was 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.

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.

4.1.2 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.

4.1.3 Preliminary 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 [aʊ] 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.

4.2 Acceptance/Rejection of Speech using Cheater Mode

Although the preliminary tests performed using syntactic error modelling for acceptance/rejection showed promise, we felt that we could not always rely on this syntactic component to perform adequately. We expected that it would be too difficult to develop an optimal syntactic model for every word spoken into the recognition system. And even though the syntactic error modelling method performed well for isolated word utterances, extending this method for use with multi-word utterances seemed particularly problematic.

Therefore, we determined that we could develop more confidence in the scores output by the decoder if the only parse being considered was the expected response. When the decoder prepares to provide text output, we would tell it beforehand only the exact words that are expected to be spoken. No other alternative parses are provided. When the decoder has only one parse for decoding, and that parse is the expected response, we term this *cheater mode.*

It was also necessary to develop a procedure to be able to easily interpret how the recognition system will behave with respect to correct acceptances, correct rejections, false alarms and incorrect rejections. A plotting scheme was designed in order to see how what percentage of the data would fall into each category. The main distinction between the data sets is scores from correctly pronounced words versus scores from incorrectly pronounced words.

Initially, we performed a comparison of the syntactic error modelling method versus the cheater mode method. We wanted to establish that cheater mode decoding would provide us with acceptable accuracy in terms of acceptance and rejection. Figure 1 shows a plot of the results of the three cases above ("TESTWORD", "Miscue in POTENTIAL_OBSERVED" and "Miscue NOT in POTENTIAL_OBSERVED"), but this time tested using cheater mode with the decoder. This plot is for the test word "any" for only one speaker, LSO. The vertical scale is the percentage of the data set. The horizontal scale is the range of utterance scores. The black squares indicate the percentage of the correctly pronounced utterances which would be *rejected* at a given score. The white squares and black diamonds mark the percentage of incorrectly pronounced utterances (miscues) which would be *accepted* at a given score. The *cross point* of these data sets indicates the optimal *acceptance/rejection threshold.* In other words, this point marks where the score is which will accept the most correctly pronounced utterances while rejecting as many incorrectly pronounced utterances as possible.

The cross point(s) on figure 1 can be compared with the results shown in the first line of table 2 above. For example, an utterance threshold value of '49' as shown in figure 1 would yield about 90% correct rejection of potential-observed words, while correctly accepting 100% of the test words. Table 2 has 94% and 90% for these values respectively.

Figure 1: Acceptance/Rejection Chart for "any" (LSO)

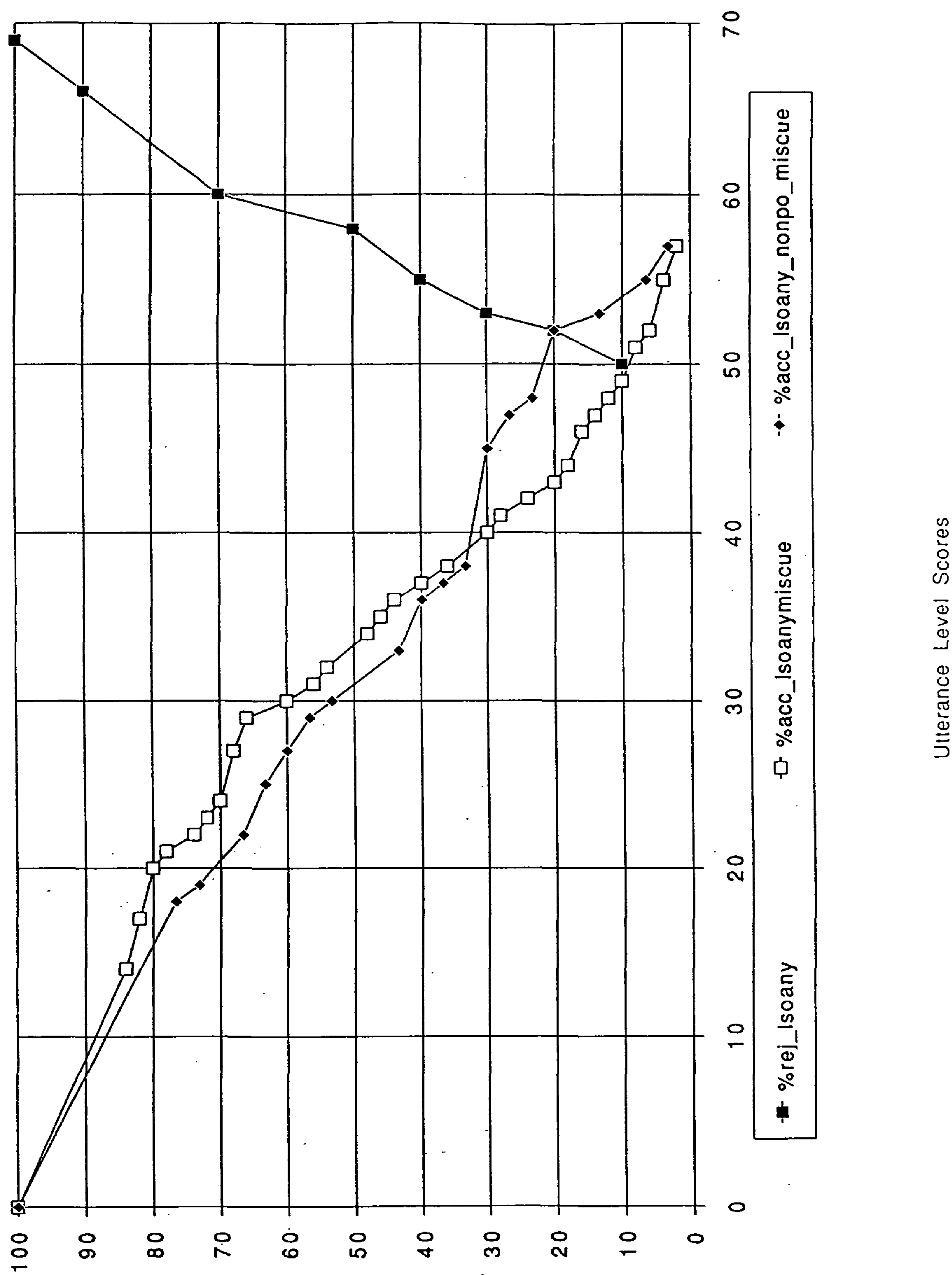
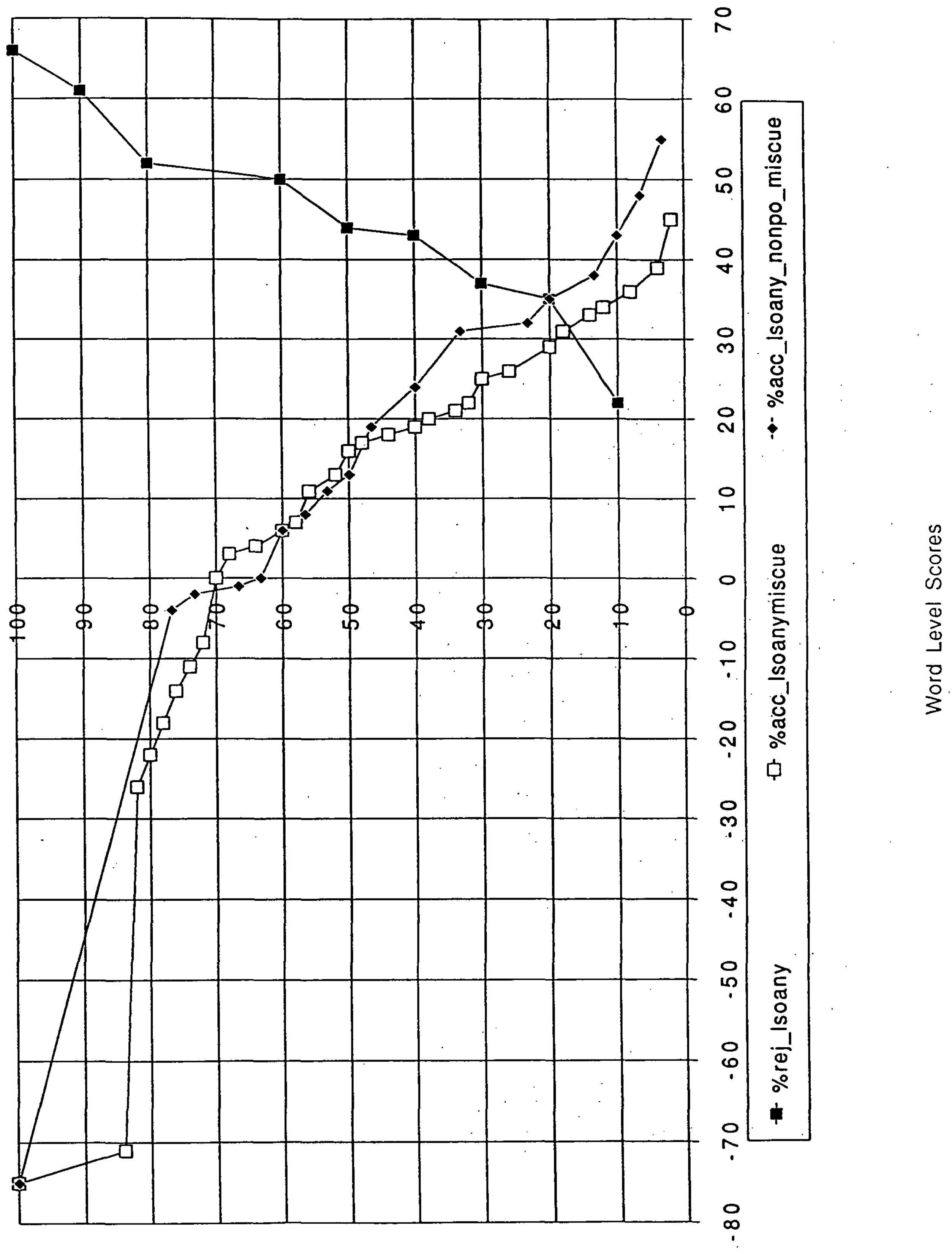


Figure 2: Acceptance/Rejection Chart for "any" (LSO)



It is also possible to examine the acceptance/rejection threshold as it pertains to the scores assigned to individual words within an utterance. In the preliminary test case, all utterances contained only one word. Therefore, an acceptance/rejection plot of the word scores should be nearly identical in form to the plot for utterance scores. Figure 2 shows that this is the case (at least for this limited data set).

Since the preliminary isolated word test only chose miscue words at random, we needed to perform a more realistic test. We wanted a set of expected response data which had a corresponding set of realistic miscues. We attempted to locate both an isolated word test set and a multi-word test set, each of which had miscues which had been culled from actual reading tests or instruction.

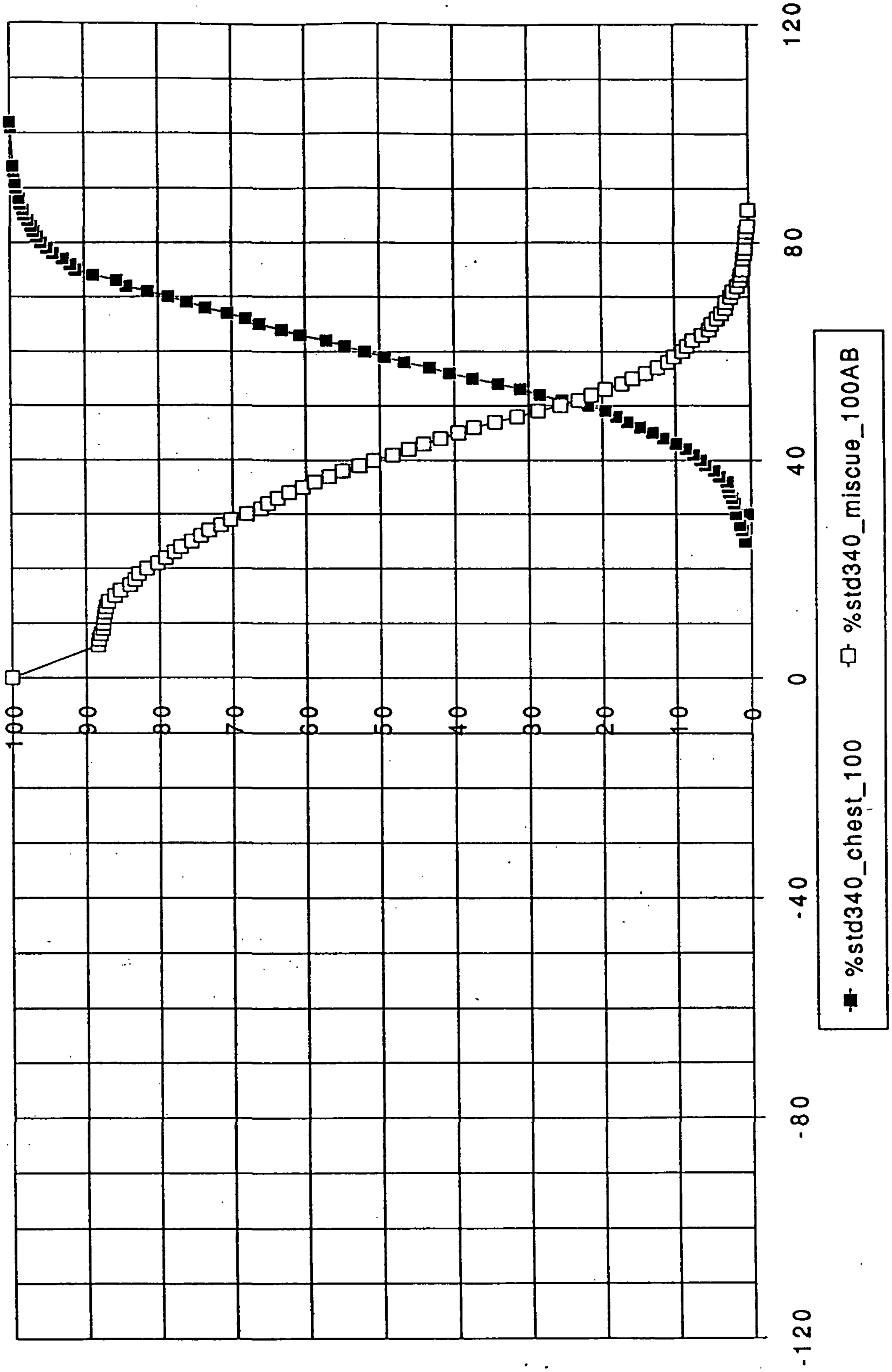
For the isolated word test, we were unable to find any published results of actual miscues encountered with readers (at any grade level). We did locate several examples of lists of isolated words to be used in reading testing or for reading exercise, but none had provided a corresponding list of known miscues. Therefore, we created our own test set of approximately 300 words. A third of the words were the expected response words. Another third were the corresponding 'close' miscue; the final third were a 'less close' set of corresponding miscues. The 'close' miscues were chosen by their relative proximity to the expected response in two categories: grapho-phonemic similarity and semantic similarity. In the isolated word case we also were making the assumption that all miscues would be considered 'word substitutions'. (For example, there were no two-word miscues for an isolated-word expected response.) The words for this test are displayed in Appendix A. Speech data for this test set was collected from six male speakers.

For multi-word testing, we located case examples of reading miscues as explained in published literature. These reading miscues were mostly from stories read by elementary school children. (No case examples of adult reading miscues were discovered in the literature.) From these case examples, we compiled a set of 67 multi-word expected response sentences. Many (but not all) sentences had one or more corresponding miscue sentences. There were a total of 67 miscue sentences. Speech data for this set of 134 sentences was collected from five male speakers to comprise the multi-word test set. The sentences for this test set are given in Appendix B.

Figure 3 shows a chart of the two data sets for the new isolated word tests. The score scale is for utterance level scores. Figure 4 displays the results for the same test set, but using word level scores. The cross points indicate that an utterance (or word) threshold near 50 would result in about 25% of the correctly pronounced being rejected, while allowing 25% of the incorrectly pronounced words to be accepted. We expect that score normalization will improve this acceptance/rejection rate.

Figure 5 shows a chart of the results for the multi-word sentences. This plots word score information for three distinct data sets: 1) black squares indicate the percentage of *correctly pronounced words in the correctly pronounced sentences* which would be rejected at a given score; 2) white squares indicate the percentage of *incorrectly spoken words in the miscue sentences* which would be accepted at a particular score; and 3) black diamonds mark the percentage of *correctly spoken words in the miscue sentences* which would be rejected at a given score. In this case, a word threshold near '37' correctly accepts about 78% of the correctly spoken words, while correctly rejecting 78% of the mispronounced words.

Figure 3: Acceptanc/Rejection Chart for Isolated Words



Utterance Level Scores

Figure 4: Acceptance/Rejection Chart for Isolated Words

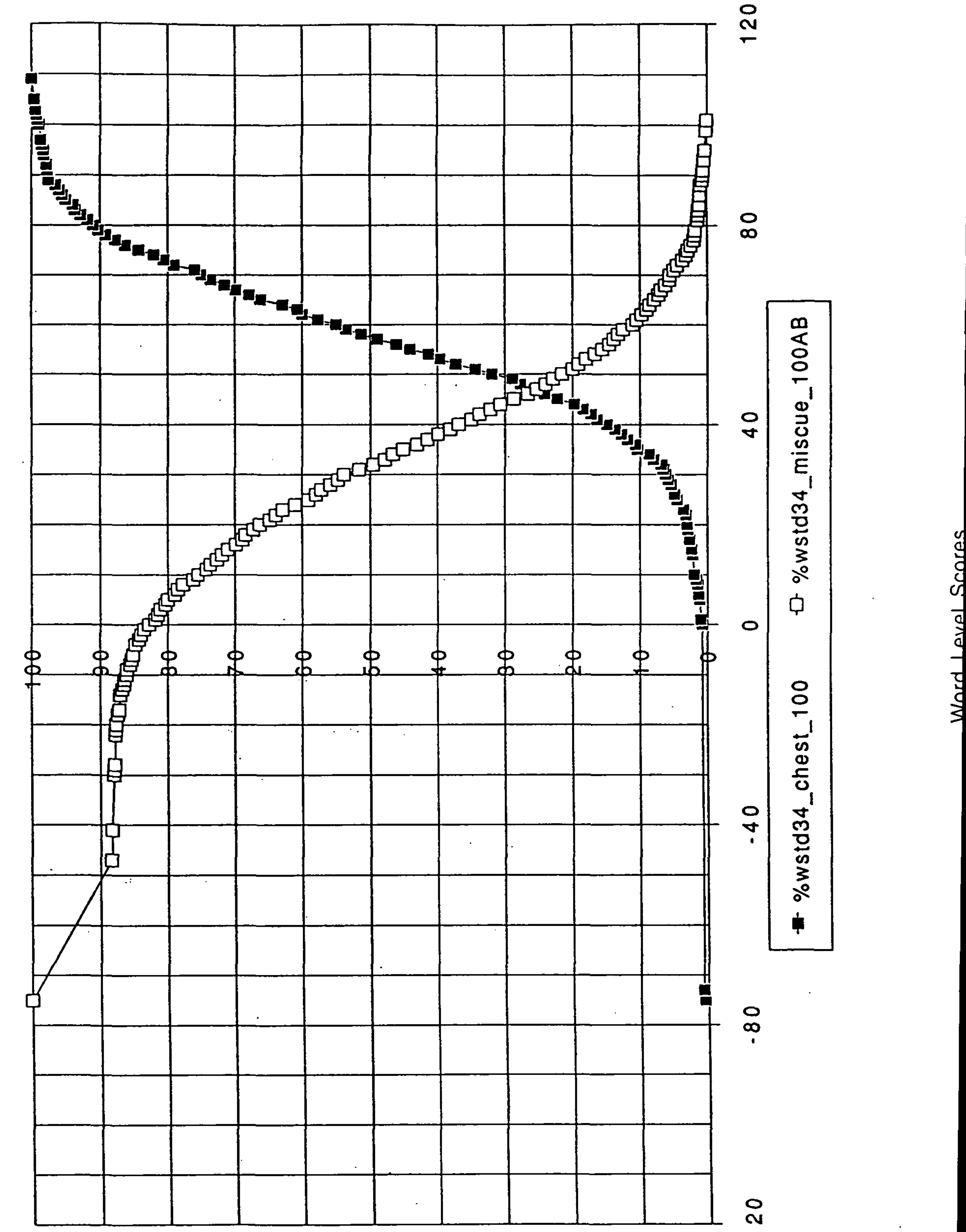
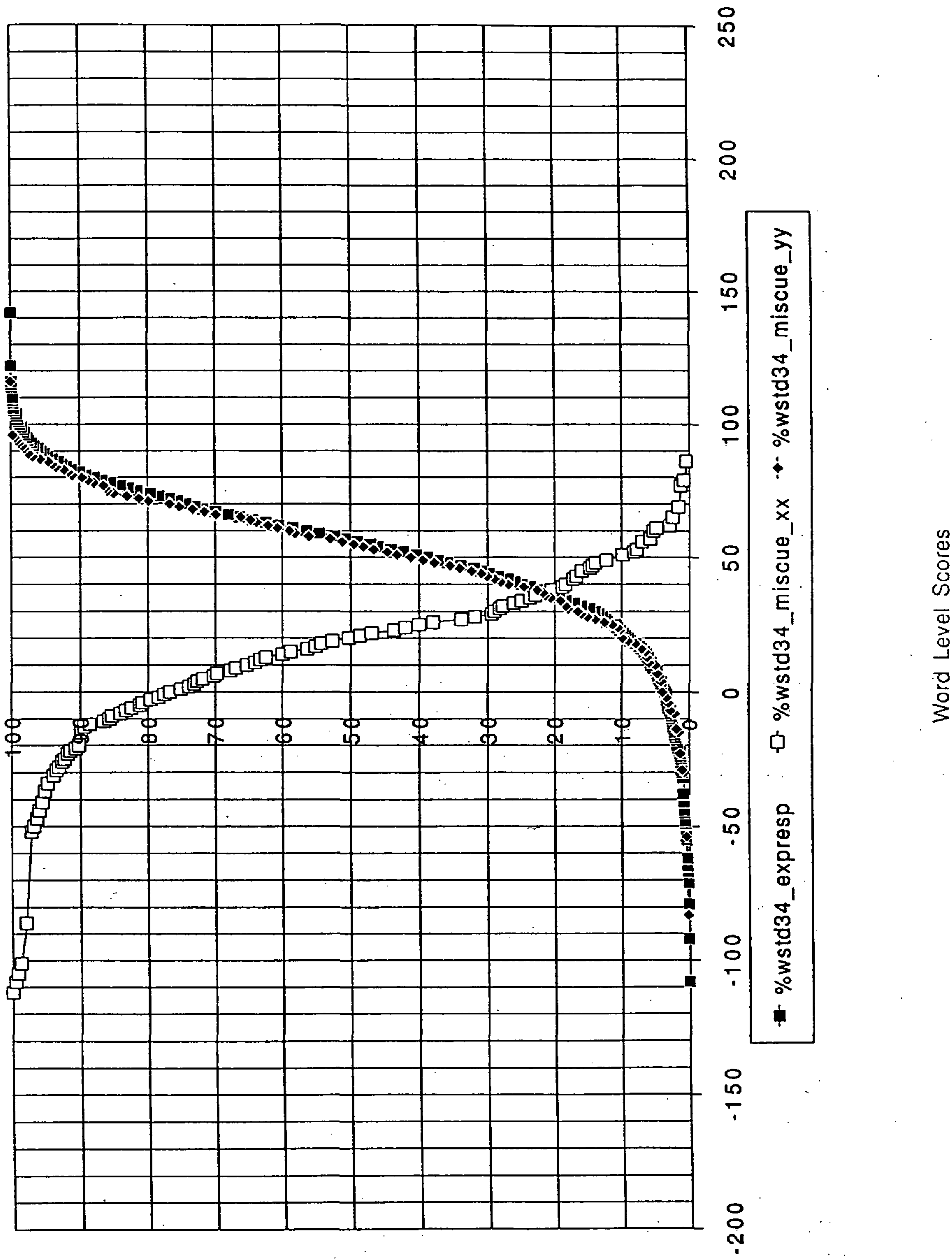


Figure 5: Acceptance/Rejection Chart for Multi-Word Utts



4.3 Score Normalization

In order to gain some improvement in the acceptance/rejection thresholds, we implemented score normalization techniques in the decoder. In addition, we performed tuning of the score normalization mechanisms in the codebook in an attempt to improve the scores after they had been normalized.

4.3.1 Results of Score Normalization

We were able to perform the score normalization effort as described above in order to produce word and utterance scores which would be normalized to a score of zero. A test was performed to compare the scores produced by the decoder in "cheater mode" before normalization, and then after normalization. Figures 4 and 5 show the acceptance/rejection plots of word level statistics for the isolated-word and multi-word test before normalization; figures 6 and 7 show the same tests after normalization has been implemented in the decoding algorithms. The main effect here is a leftward shift in the optimal acceptance/rejection threshold to the region around '0' in the horizontal axes. This indicates that the normalization techniques are successful.

4.3.2 Results of Score Normalization Tuning

The expected phonetic code scores are used as the initial values for doing score normalization. The expected scores associated with the phonetic codes were determined from populations of the training data. This is the source of the initial scores of the phonetic codebook also.

A process of adjusting the scores of the codebook for improved accuracy is used to further prepare the codebook for delivery. An analogous method of adjusting the normalization scores for phonetic codes has been implemented to improve the normalization performance. The adjustment is accomplished by making slight modifications to the normalization scores and the codebook scores whenever they produce the wrong result, i.e., whenever the score of an incorrect decoding is higher than the normalization score, or when the score of a correct decoding is lower than the normalization score. If the normalization score classified the decoded result correctly, no adjustment is made.

We have called this method codebook tuning when applied to the codebook alone, and normalization tuning when applied to the codebook and the normalization scores together. The method used employs the Perceptron learning algorithm. The normalization tuning method extends the codebook tuning approach by using a threshold which is determined from the sum of the phonetic code specific thresholds (the normalization score offset values). This has the effect of training the normalization scores together with the codebook scores, so they should improve together.

Figures 8 and 9 show the acceptance/rejection results after normalization tuning has been applied to the phonetic codebook. Figure 8 contains the results of the isolated word test; figure 9 shows the results when tested using multi-word sentences. The figures show that the word scores are better normalized in that the acceptance/rejection cross point is located at zero on the horizontal (score) axes. The cross points also appear to be slightly lower on the vertical axis (than in non-normalized tests), indicating an improvement in the acceptance/rejection threshold. One other aspect of the normalized codebook is that it tends to cause more bad parses. This results in better rejection of mispronounced sentences. In

Figure 6: Acceptance/Rejection Chart for Isolated Words

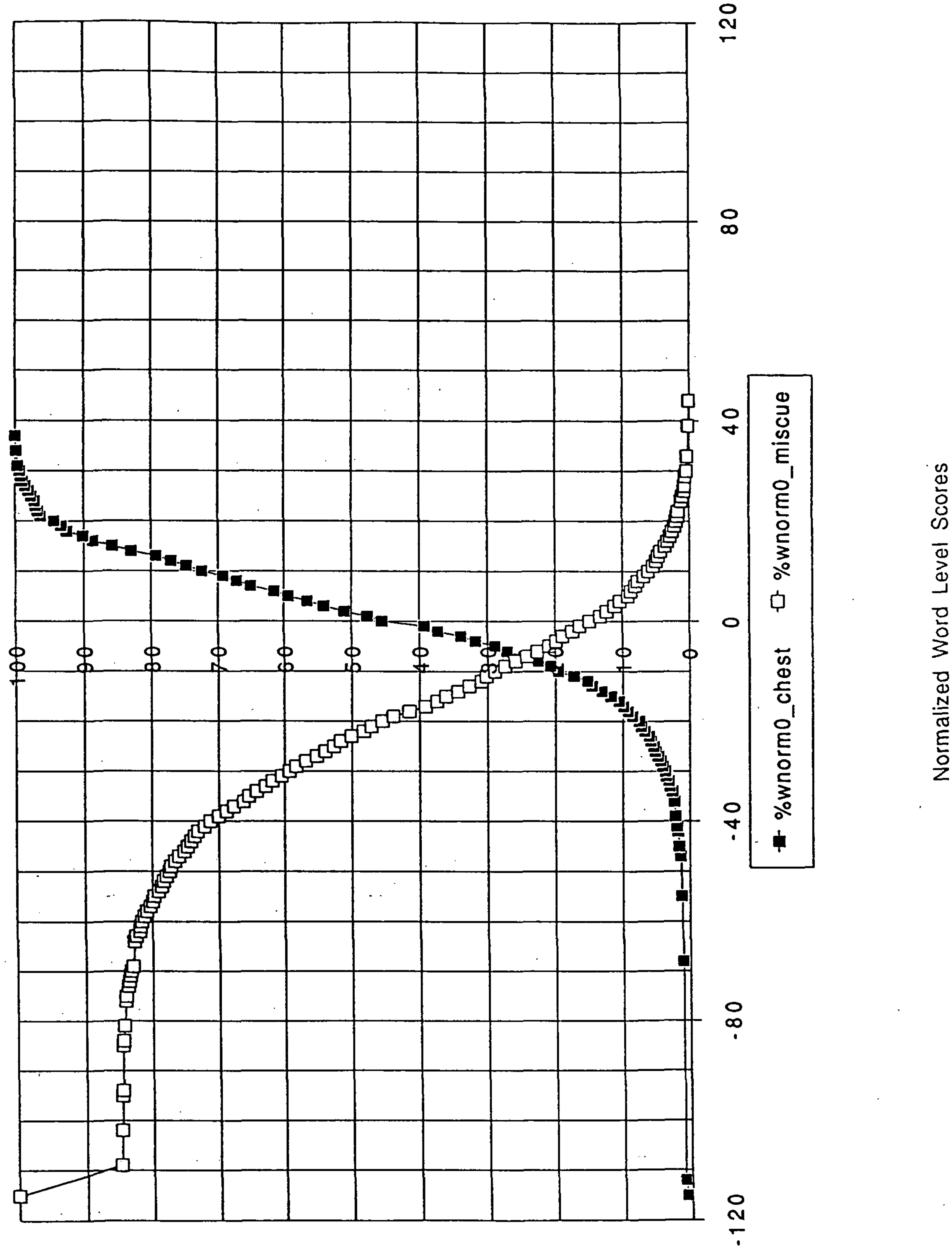


Figure 7: Acceptance/Rejection Chart for Multi-word Utts

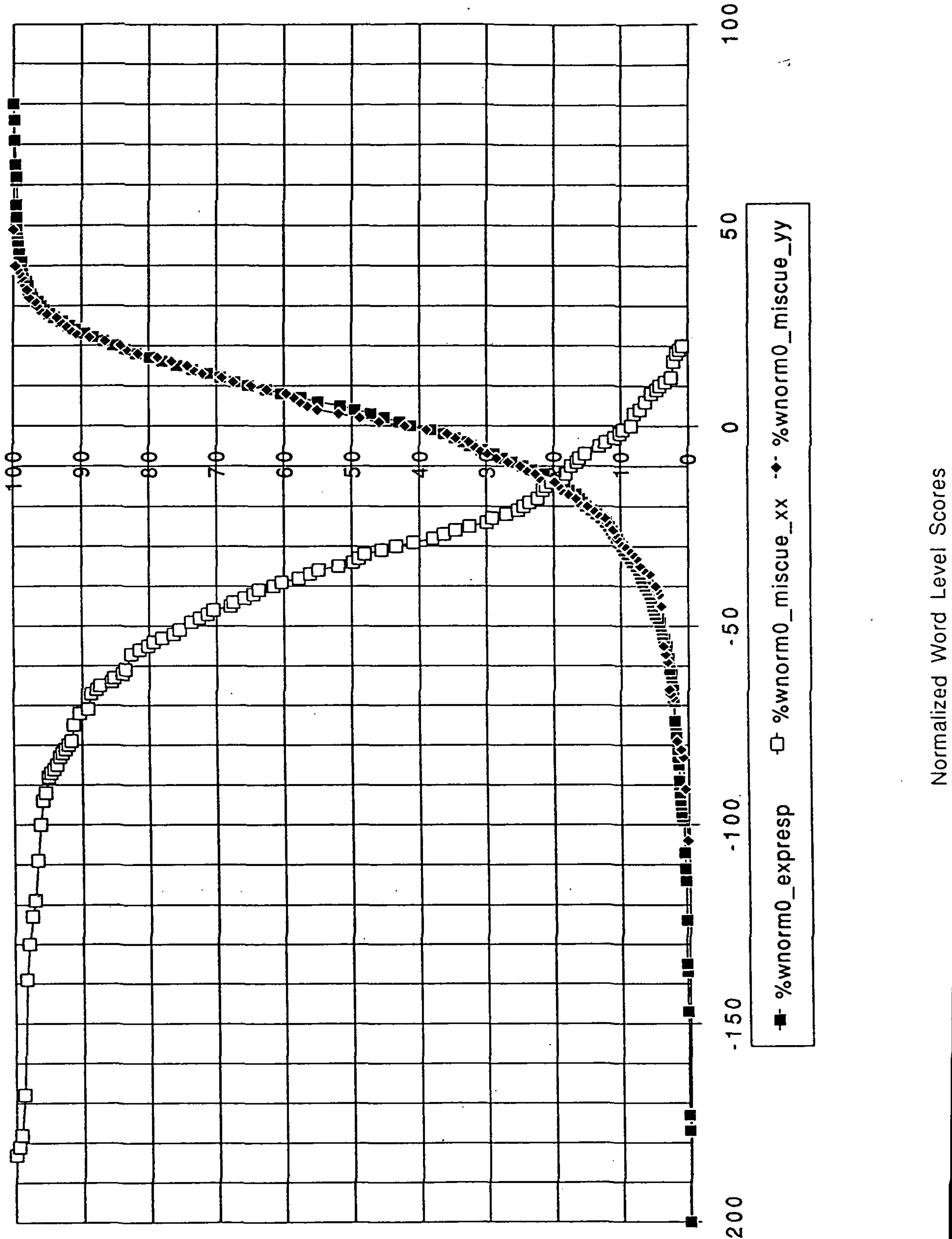


Figure 8: Acceptance/Rejection of Isolated Words

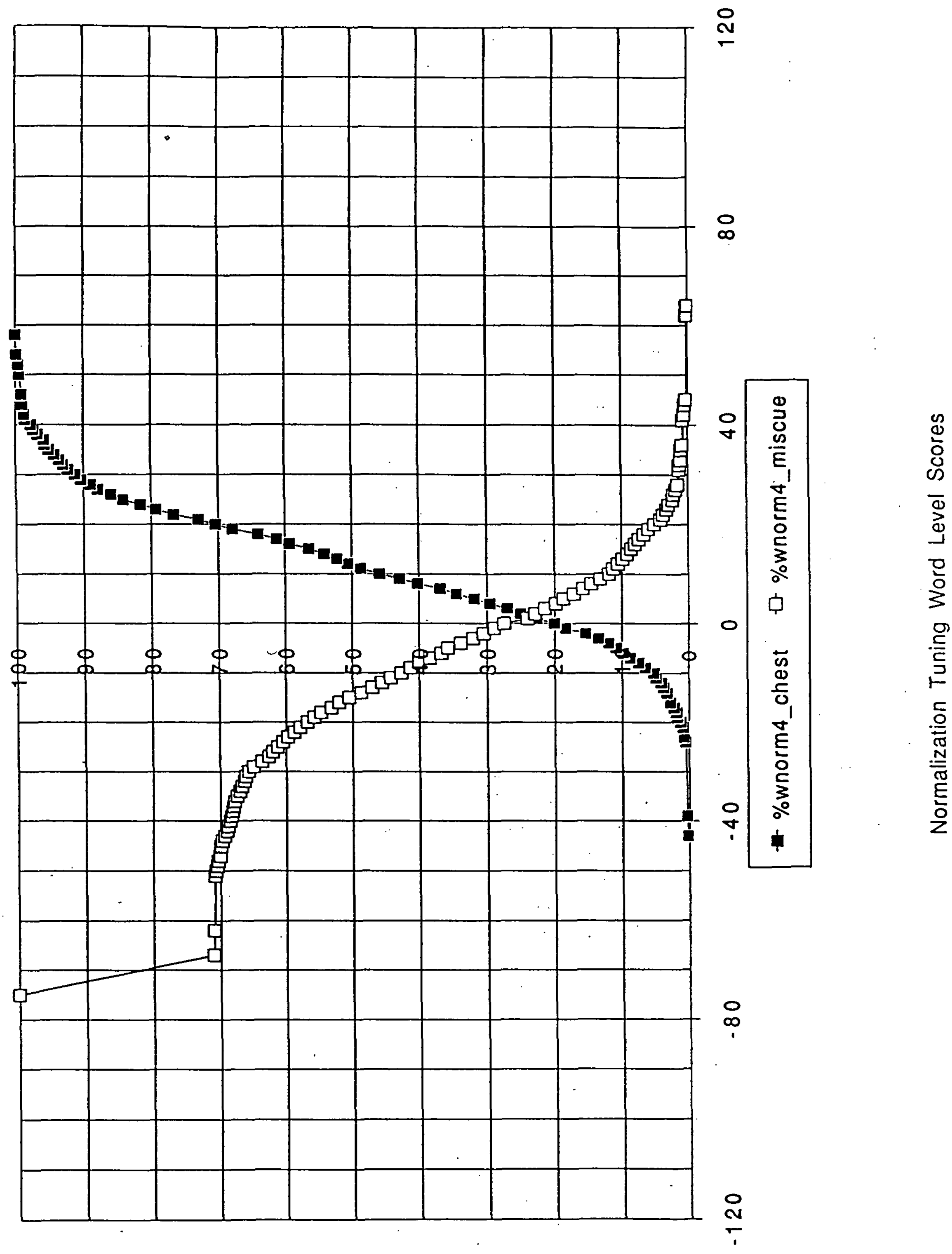
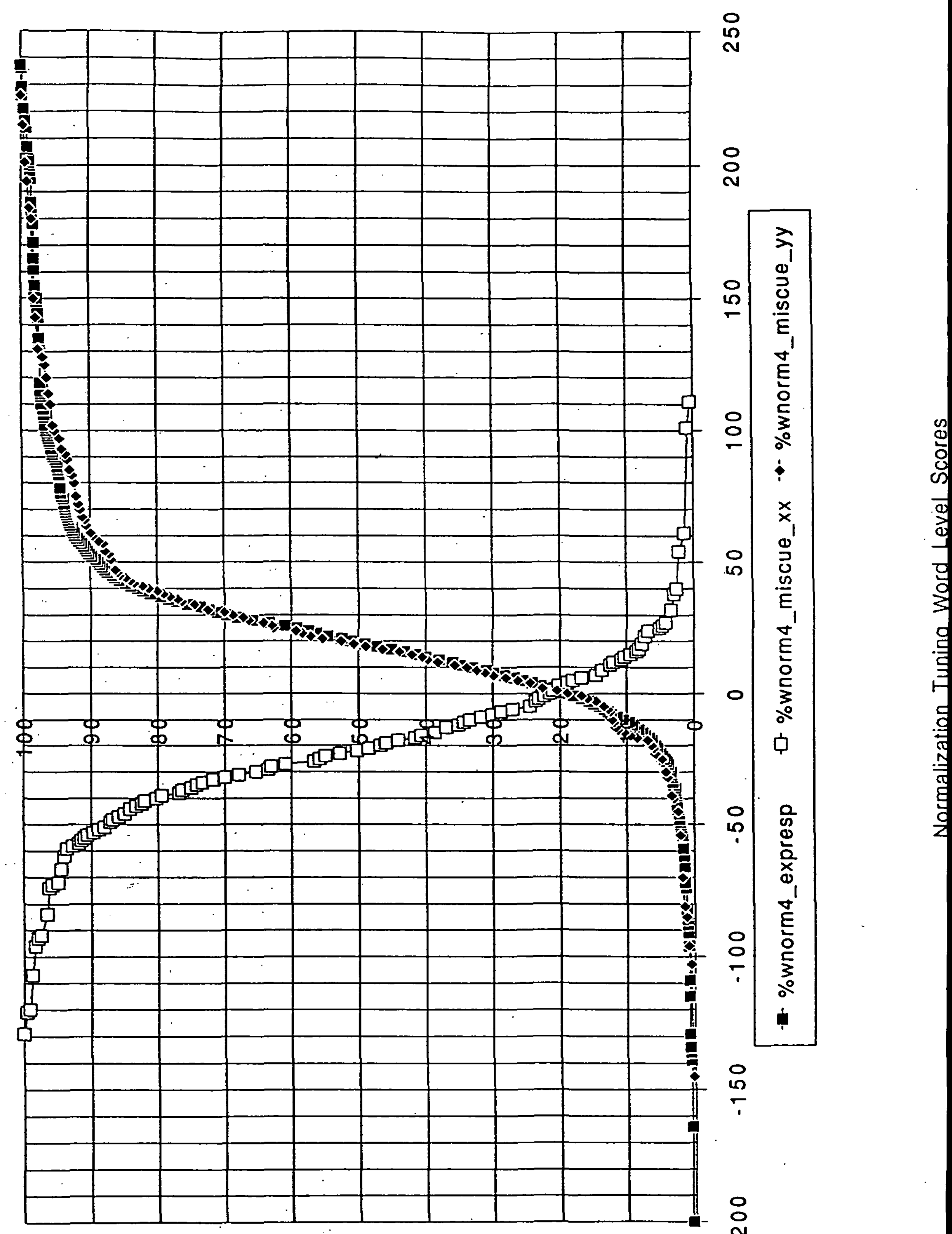


Figure 9: Acceptance/Rejection Chart for Multi-word Utts



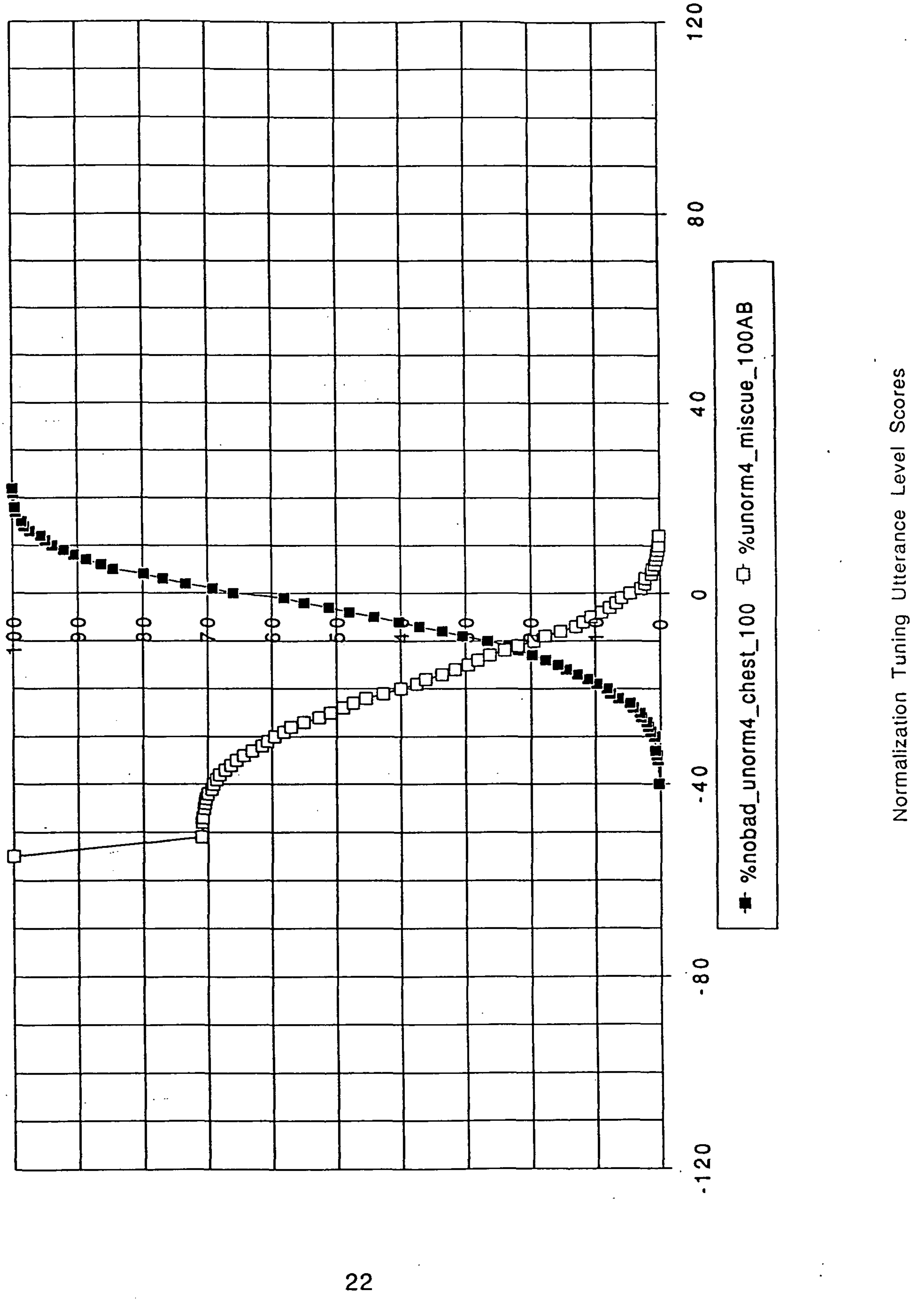


Figure 10: Acceptance/Rejection Chart for Isolated Words

figure 8 this can be observed in the upper left corner of the chart. The upward jump at the left-most data point represents the percentage of mispronounced words which resulted in bad parse rejections. In this test, these bad parses constitute nearly 30% of the mispronounced data.

In the isolated word case, it seems that the utterance score can be used to achieve a better acceptance/rejection threshold. This can be seen in figure 10 when compared with figure 8. In figure 10, the cross point appears lower and the area is smaller in the triangle-shape under the cross point.

4.4 Phoneme Error Modelling

We experimented with 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:

$$
\begin{aligned}
\text{1-Syllable} = \; & (\{ G_i \mid C_i \mid K_i \}) \\
& \{ V_o \; (K_f) \\
& \mid V_y \; G_y \; (K_y) \\
& \mid V_{wy} \; \{ G_w \; (K_w) \mid G_y \; (K_y) \} \\
& \}
\end{aligned}
$$

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.

This one-syllable meta-words syntax was extended to account for multi-syllable words (and phrases) by allowing the syntax to be iterative with respect to the meta-syllable. We also wanted to test this metaword syntax to see how well it covered the English language. We were able to do this by creating a program which converts words to phoneme-words using a phonetic dictionary. We tested words and sentences of English, for a total of 3611 unique vocabulary items. As a result of this test, several changes were made to the metaword syntax to allow a more complete coverage of the phonotactics of English words. We were also forced to create some non-legal phonotactic rules in order to allow for some oddities in the phonetic spellings contained in our phonetic dictionary.

The resulting phrase rule structure for the metaword syntax is as follows:

```
S ->  INIT_SYLLABLE_  ({* NEXT_SYLLABLE_ *})

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

NEXT_SYLLABLE_  -> (q_) ( { Gi | Ci | Ki | FLAP_ } )
              { Vo (Kf)
              | Vy { Gy(Ky) | R_ALL }
                | 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_

Gi == l_ r_ w_ y_

Gw == w_

Gy == y_

FLAP_ == tt_
```

```
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_
|{cx_|sh_} r_
|s_{f_|l_|m_|n_|w_|bx_|dx_|gx_}
}

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

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

Kf -> { K_LRf | K_STOPf | K_FRICf }

K_LRf -> { LONLY_ | RONLY_ | LRBOTH_ }

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

RONLY_ -> r_ (RONLY_CLUSTERS)

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

LRBOTH_ -> {l_|r_} LRBOTH_CLUSTERS

LRBOTH_CLUSTERS -> {{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_}
```

```
R_ALL -> r_ ({ RONLY_CLUSTERS | LRBOTH_CLUSTERS })

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_)
| ng_ th_ (s_) }

K_STOPf ->
{ m_ ({dx_|{px_|f_}({s_|tx_})|z_|px_ f_})
| n_ ({dx_(z_)|th_({s_|tx_})|s_(tx_)|sh_(tx_)|zh_(dx_)|tx_(s_)|z_})
| ng_ ({dx_|gx_|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.

The metaword syntax can also generate/accept multiple-syllable words, as in these examples:

```
twizlyor:   tx_ w_ agrave y_ z_ l_ y_ ograve r_
nyawoilegg: n_ y_ agrave w_ ograve y_ l_ eacute ng_
cleltallyo: kx_ l_ enoth l_ q_ aacute l_ l_ y_ oacute
orpya:      ograve r_ px_ y_ eschwa
tragthrok:  cx_ r_ anoth gx_ th_ r_ oacute kx_
nyoakdeng:  n_ y_ onoth kx_ q_ dx_ w_ eschwa ng_
broar-ain:  bx_ r_ ograve r_ q_ eacute n_
awbraw:     agrave w_ bx_ r_ agrave w_
```

We used a set of speech data from 4 male speakers who provided about 200 isolated words each. Results on one-syllable words appeared encouraging. However, phoneme accuracy dropped significantly when attempting to use the metaword syntax to decode multiple-syllable words. The phoneme accuracy for the multiple-syllable word test set was 53.13%; word recognition (i.e. getting all the phonemes correct for a given word) was only 3.97%.

We designed a method whereby we could tune the phonetic codebook to be better at decoding the phonemes. After six iterations of tuning, this resulted in some improvement, the best of which occurs with the fourth iteration (as shown below).

ITERATION	PHON.ACC.	WORD ACC.
0	53.13	3.97
1	55.85	4.63
2	57.06	4.89
3	57.38	4.50
4	57.89	4.89
5	57.52	4.89
6	56.42	4.76

For the 4th iteration codebook, we examined the performance of several subsets of the data divided into groups by syllable counts. One-syllable words perform noticeably better.

SYLLABLES	PHON.ACC.	WORD ACC.
1	65.02	17.14
2	60.09	2.53
3	56.28	0.53

We can improve performance only slightly by re-decoding each of these subsets of data using a different syntax. For the one-syllable words, we use a syntax that allows one and only one syllable. For the two-syllable words, we use a syntax that allows exactly two syllables, etc.

SYLLABLES	PHON.ACC.	WORD ACC.
1	66.50	17.71
2	60.65	2.53
3	57.70	1.59

Even though these results may not seem very exciting, it may be that they are somewhat deceiving with respect to how well the recognizer is performing. When we examine the phoneme errors that are made, we see that many of the errors are close substitution errors. For example, the word *pole* is converted into phoneme-words as *px_ oacute l_*. The recognizer provides the following output for this word: *bx_ oacute l_*. This is actually very accurate, even though the "phoneme accuracy" is only 66%. If we interpret each phoneme into a set of distinctive features, a numerically calculated accuracy of *bx_ oacute l_* would be much higher. Even if we consider an extremely simple "3-feature" representation of each phoneme such as *place, manner* and *voicing*, we can see that getting *bx_ oacute l_* for the word *pole* would be 88% accurate rather than 66%. This type of consideration would also be feasible for the literacy tutor application since a mechanism may need to be developed to provide "friendly" feedback, instead of simple phonetic feedback which may be uninterpretable by students. (e.g. A student may make better progress by receiving the feedback "Say it again, this time popping the 'p' more", rather than "Say it again, this time more like 'p' than 'b'.")

4.5 Demonstration Software

In the spring of 1991, we supplied a preliminary version of the 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.

For the final deliverable, we have enhanced this application. It now allows a sentence to be sent in to the serial port. This sentence is the expected response. The application will use the new score normalization techniques and tuned codebook (*norm.prf*), along with the acceptance/rejection thresholds to notify when a word or sentence has been poorly spoken. For phoneme level feedback, we are delivering the metaword syntax and the speaker model (*meta.prf*) that has been tuned for use with the metaword grammar. The metaword could easily be incorporated into the demonstration application to do "second-pass" decoding or "re-prompting" of isolated words after they have been "rejected" by the acceptance/rejection threshold. The best phoneme output for that word could then be sent to a module to determine appropriate user feedback and pronunciation instruction. The application can use the codebook switching mechanisms in the Phonetic Decoder Interface to enhance performance for sequential decoding, using the normalized speaker model in the acceptance/rejection phase, and the metaword speaker model for decoding to the phoneme level.

5. Conclusion

The research performed for this project was successful in providing better mechanisms for using speech recognition in a literacy tutor application. Using a combination of scoring normalization techniques and cheater-mode decoding, we are able to provide a reasonable acceptance/rejection threshold. In continuous speech, the system has been tested to be able to provide above 80% correct acceptance of words, while correctly rejecting over 80% of incorrectly pronounced words.

Acknowledgements

Special thanks to the following employees at Speech Systems Incorporated for their support on this project: Mark Anikst, Robbie Ishihara, Olivia Pangerapan and John Turley.

Appendix A: Isolated-Word Test Sentences

Note: The expected response is in the first column. The corresponding 'close' miscue is in the second column, with the 'less close' miscue in the third column.

ER:	*Miscue A:*	*Miscue B:*
foreign	forward	for
midline	middle	outlined
advanced	absence	branch
scattered	scarring	pattern
cervix	certain	curved
adult	occult	dual
structure	structures	fracture
widespread	wide	withdrawal
kerley	clearly	early
larger	largest	longer
hodgkin's	rhonchi	margins
exact	effect	exams
suggest	suspect	shaggy
femur	feature	further
straightened	stretching	right
cystic	cysts	blastic
patchy	patch	partial
borders	border	ordered
given	seven	even
defect	effect	detect
trapping	capping	rapid
clothing	clearing	ankylosing
unchanged	change	chains
oblique	opaque	oblong
tonsil	tension	senile
older	old	shoulder
looking	marking	leaking
mildly	mild	midline
chest	crest	breast
twelfth	twelve	tenth
film	films	field
rods	rod	ards
brain	drain	membrane
of	if	from
see	seen	cell
loss	less	mass
slight	light	spite
ninth	nine	month
eighth	eight	eggshell
widths	width	with
be	been	tree
haze	has	hazy
edge	edges	wedge
if	it	its
its	it	if
bulge	bulging	bulb

appear	appears	appearing
crowding	crowded	caudad
obtained	retained	obscured
shortness	short	sharply
smaller	smallest	small
beyond	behind	below
setting	settings	section
fingers	finger	fungal
central	ventral	senile
wiring	wire	wires
seventh	seven	tenth
along	long	oblong
outlined	outside	midline
contoured	contours	contour
moving	missing	going
routine	retained	round
certain	current	series
them	then	they
is	its	sized
weeks	week	weight
patch	patches	branch
leg	legs	long
shaped	shape	sharp
huge	large	high
days	day	today's
walls	wall	well
new	few	old
round	around	rounded
rest	best	crest
cyst	cysts	cystic
gross	grossly	across
most	almost	post
anterior	arterial	antrum
hematoma	hemangioma	heart
abdominal	duodenal	stomach
thoracotomy	thoracotomies	colectomy
bronchovascular	bronchus	infraclavicular
appreciable	appears	stable
macrocalcification	macrocalcifications	microcalcification
pneumoperitoneum	pneumopericardium	pneumonia
atherosclerotic	atherosclerosis	sclerotic
column	colon	cavum
improved	increased	worsened
exams	exam	exact
define	defined	fine
aspects	aspect	effect
recess	recent	assess
hickman	thick	hancock
greater	gutter	greatest
retained	remain	obtained
margin	margins	marking
fullness	full	filling
also	all	almost
very	there	were

Appendix B: Multi-Word Test Sentences

Note: Expected response sentence are preceded by "ER:". Sentences that follow an ER sentence are miscue sentences for that expected response.

ER: joe was not happy

ER: he wanted a pet
he's asked a pet

ER: he found a pet
he for a pet

ER: it was a goat
is was a

ER: mom said no

ER: dad said no no no

ER: no goat for joe
no not for john

ER: joe heard a sound
john was sad

ER: he looked around
he looked he looked

ER: it was a dog

ER: a little dog

ER: joe patted the dog
john patted the dog

ER: hello joe said
hello john said

ER: i like you

ER: bow-wow
boo-hoo said the dog

ER: nice dog said joe
dog said john

ER: come with me

ER: be my dog

ER: be my pet

ER: at last uncle bill and harriet and mom and dad were at home
at last bill and hi and moo and dad we were at home
at last bill and hi hay her and moo and dad we were at home

ER: harriet said i'll go to see my friends
hare said i'll go to see my friend

ER: all of them have missed me so much
i'll all of them have made some me so much

ER: i know they all will want to see me
i know that i'll what to see me

ER: that's a very good idea said mom
that's a very good answer said moo

ER: why don't you go see all of them

ER: harriet ran to pat's house
hare are ride to pick home to pick's home
hare ride to pick home
hare are to pick home

ER: she ran up to the door
she ride up to the door

ER: pat's mom came to see her
put puts moo come to see her

ER: is pat home asked harriet
its put home answered hare
its is it is put home answered hare

ER: he will want to see me

ER: i have been away

ER: no said pat's mom
no said pat moo

ER: pat is not here
pat said not here

ER: he is away right now
he is away wry now right now

ER: but he didn't say he missed you
but he didn't say he must you

ER: i can get to the park fast
i can get ahead to the park first faster
i can get ahead to the park first
i can get ahead to the park faster

ER: you and i will run
you and i will race

ER: we will run to the park
we will race to the park

ER: i can run fast and you can't
i can run fast you can't

ER: turtle said you can run fast rabbit and i can't
turtle said you can run fast rabbit but i can't

ER: that's why i got here first
that's how i got here first

ER: you run fast rabbit
you can run fast rabbit
you are fast rabbit

ER: but you stop and i don't
but you stopped and i didn't

ER: you can't run turtle
i can't run turtle
you can run turtle
you can't run [fA"] turtle

ER: we will run to the park
i will run to the park

ER: i'll fix a light and drop it to you
i'll fix the light and drop it to you

ER: mrs. miller had gone to visit a neighbor
mrs. miller had gone to visit the neighbor

ER: it was safely outlined in a library book
it was safely outlined in the library book

ER: in one corner and along one side
in the corner and along one side
in one corner and along the side
in the corner and along the side

ER: he was hanging up the two telephones
he was hanging up two telephones
he was hanging up the telephones

ER: then one afternoon he left
then the afternoon he left

ER: one picture showed a large black crow
a picture showed a large black crow

ER: he printed them upstairs in his darkroom
he printed them upstairs in the darkroom

ER: after the cut in his allowance
after the cut in the allowance

ER: he hurried to his cellar work table
he hurried to the cellar work table

ER: it was in his toolbox
it was in the toolbox

ER: the lady led me toward his office
the lady led me toward the office

ER: she was playing with the camera
she was playing with her camera
she was playing with a camera

ER: he was always like one of the uncles
he was always like one of his uncles

ER: he was a real chemist with a company in switzerland
he was a real chemist with his company in switzerland

ER: she came to the house
she came to our house

ER: he wagged a finger at andrew
he wagged his finger at andrew

ER: he was grabbing for the finger
he was grabbing for his finger
he was grabbing for your finger
he was grabbing for a finger

ER: they were pictures of their father
they were pictures of the father

ER: they took pictures of their mother
they took pictures of the mother

ER: that's mine said my baby brother
that's mine said the baby brother