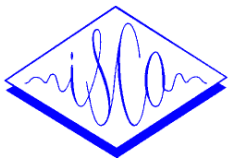

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

MULTIPLE-LEVEL EVALUATION OF SPEECH RECOGNITION SYSTEMS

John F. Pitrelli, David Lubensky, Benjamin Chigier, and Hong C. Leung

Speech Technology Group, Artificial Intelligence Laboratory
NYNEX Science & Technology, Inc.
500 Westchester Ave., White Plains, NY 10604, U.S.A.

ABSTRACT

Evaluations of speech recognizers typically focus on somewhat idealized versions of the types of utterances the recognizer would confront in a real application. One issue which our group has discussed previously is the use of laboratory speech rather than real-user speech, and the resulting over-optimistic projections of performance. This paper focuses on another problem: typical evaluations often exclude some or all classes of utterances which would occur in a real application but do not precisely match the type of input for which the recognizer was designed. Some example categories are utterances which include excess words not in the recognizer's vocabulary (*non-target* speech), utterances which lack target speech, and "utterances" lacking any speech at all. Such inputs to a recognizer may result from non-compliant users, or from pre-processing errors such as imperfect end-pointing. We propose a more comprehensive evaluation strategy, using as an example an evaluation of a recognition system prototype for a city-name-recognition application. Our strategy is designed to meet two goals — to evaluate automation potential realistically, and to provide diagnostic information to pinpoint directions for future work on the system. To these ends, our evaluation treats both the overall system and the individual component modules within it. We learn that a surprisingly wide variety of "recognition rates" can meaningfully describe the accuracy of a system or portions of it. Consequently, accuracy statistics must be interpreted and/or compared with extreme care.

1. INTRODUCTION

To operate effectively in a real-world setting, a speech recognition system must be able to deal appropriately with inputs the recognizer was not intended to handle. Our real-user data collections demonstrate that such "non-compliant utterances" are common. In one experiment comparing prompts designed to elicit an utterance containing just a city name, the most successful prompts still failed to do so more than one-third of the time [5]. The *target* response, in our case just a city name, is sometimes embedded in a longer utterance, for example, "in *Boston* please", or even, "yes um oh I think he's in *Arlington* I'd like the number for John Smith". As a result, a recognizer needs some form of word-spotting capability to extract a target word from a longer utterance [6]. Other utterances lack a city name altogether, for example, "yes information I'm looking for the bank in Central Square", necessitating a rejection capability to decide when the recognizer's output should or should not be accepted. Because rejection algorithms are imperfect, they introduce both false acceptance and false rejection errors. Great variability in utterance duration, and our desire not to make customers wait for a lengthy time-out, require the use of end-point detection. This introduces yet another source of error into the recognition process. Thus, real-user considerations imply that evaluation of an application-oriented recognition *system* will differ significantly from the evaluation of the recognizer at its core, and for this reason a multiple-level scheme is needed to analyze the system fully.

We illustrate our evaluation scheme using a city-name-recognition system configured for partial automation of directory assistance. A block diagram of this application appears in Figure 1. The goal of the application is to prompt the user to say a city name in isolation, recognize it automatically, select that city's portion of the directory assistance database for the operator and then pass on the remainder of the call to be completed by the operator as usual.

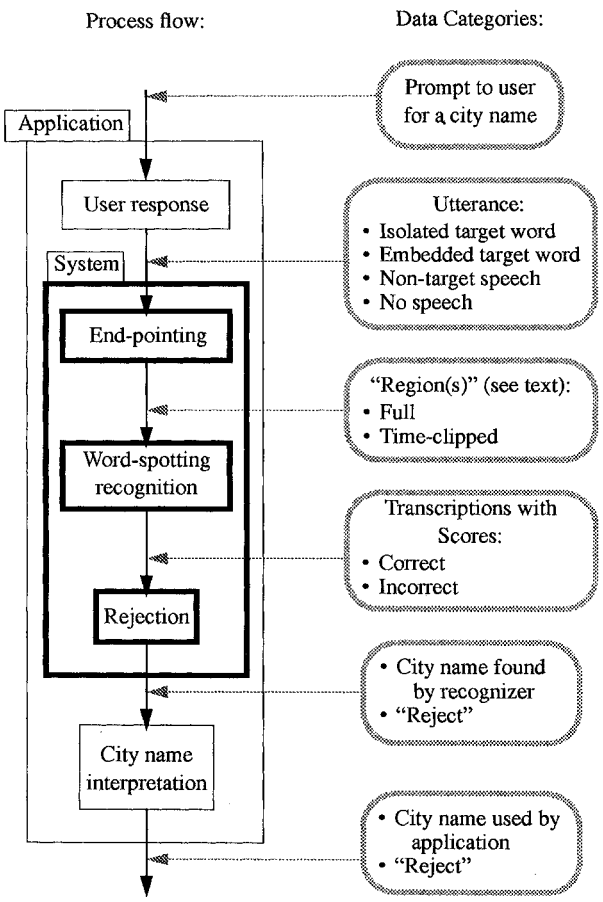

Figure 1: City-name-recognition application. Bold rectangles indicate components to be evaluated here.

The recognizer on which this system is built is a word-spotting version of the Stochastic Explicit-Segment Modeling recognizer developed at MIT and described by Leung *et al.* [3]. The recognier currently uses a vocabulary of 25 city names from the Boston area. The rejection algorithm is based on a mixture-Gaussian classifier described by Chigier [1], and the end-pointing module was developed in our laboratory.

"User response" represents human behavior rather than system performance, and so is not treated here in detail but is described for this application by Spitz *et al.* [5]. "City name interpretation" refers to the translation of the city name spoken by the user into a city name used

by directory assistance. For example, someone requesting the Allston section of Boston may say "Allston", and so the recognizer must be prepared to recognize "Allston", but the city name passed to the application should be "Boston". City name interpretation is indicative of the geography of the Boston area and the structure of the directory assistance database, rather than speech recognition, and so is also not analyzed here. However, we illustrate it to emphasize that for predictions of application-specific automation rates and accuracy, an *application evaluation* is needed to reach results like "Using prompt A, X% of user utterances lead to correct automation."

We analyze any component largely in terms of the relationship between its input categories and output statistics. These categorizations include levels of user compliance reflected in the utterance[1]; different ways that the end-pointing module may cut the utterance into *regions* of speech bounded by silence; different levels of accuracy of output of the recognizer; and whether the recognizer's output was correctly or falsely accepted or rejected.

In this paper we describe five components of our evaluation. Three individual *module evaluations* describe performance of the end-pointing scheme, the word-spotting recognizer, and the rejection post-processor. The fourth is the *system evaluation,* relating categories of input to the system to categories of system output, for example, "Y% of non-target-speech inputs were rejected." Finally, we provide an example of a *module interaction evaluation*, exploring the dependence of the accuracy of the rejection scheme on the performance of the recognition module.

2. DATABASE

This study uses a database of utterances recorded from actual users of directory assistance in the Boston area responding to our automated prompts to say a city name. It is therefore reasonable to presume that every utterance was produced by a different speaker. Recording began at the end of the prompt to the user, and ended when an operator judged that the user had finished talking. As is to be expected of real-user telephone utterances, these recordings include an appropriate sampling of background noises (background speech, radios, machinery, pets, traffic, etc.) and network-transmission effects (including band limitation, clicks, echo, ringing, crosstalk, and other noise). It should be noted that "no-speech" utterances are often far from silent because of these effects. We also note that the recording hardware is the actual configuration for the application. Thus, the data acquisition environment — background and recording configuration — is representative of the overall application condition [2].

Our database is divided into separate training and testing portions, each of which is balanced to include the correct frequency distributions of vocabulary items and utterance types (isolated, embedded, etc.) for the application. The training portion consists of 16,916 utterances. All are used to develop the end-pointing algorithm and to train the rejection classifier, and the 5798 which contain isolated city names in our recognizer's vocabulary are used to train the recognition module. For testing we use a subset of the testing portion of the database consisting of all utterances collected in response to our best prompts. This test set consists of 1901 utterances, of which 1054 contain a city name in the recognizer's vocabulary. More information on this database is provided by Spitz *et al.* [5].

3. MODULE EVALUATIONS

End-pointing

The primary task of the end-pointing module is to decide when the user has stopped talking, so that it may then direct the system to complete processing on the region of speech accumulated before that time. A secondary function is to eliminate many "utterances" which lack speech altogether.

Our end-pointing algorithm is a simple state machine which detects changes in energy in appropriate frequency ranges to determine when speech begins and ends. Before it will divide an utterance into multiple regions, the end-pointing algorithm requires a long silence between regions, so as a result it usually finds just one region in an utterance. Occasionally it finds more than one region, if there is a long pause within the utterance, and it rarely decides there is no speech, except in non-speech "utterances".

Our main concern is that the city-name portions of utterances not be eliminated, time-clipped, or divided among multiple regions sent separately to the recognizer. Accordingly, the end-pointing algorithm is biased strongly to prefer "padding" to clipping, that is, including some adjacent telephone-network noise. This is not considered a malfunction because the subsequent recognition process performs word spotting.

We evaluate the end-pointing algorithm using hand-labeled begin and end times for city names and for speech in general. We consider four categories of result. From worst category to best, they are: (1) missed the speech entirely, (2) divided the speech into separate regions, (3) time-clipping (region starts late or ends early by at least 20 ms), (4) accurate endpointing (allowing padding). Results are summarized in Table 1, with each utterance labeled according to the worst result category that applies to it. It is interesting to note that the non-target speech is more likely to be missed or clipped than the city names. This is consistent with the result of Silverman *et al.* [4] that when users answer a prompt for only a city name with extraneous words, they usually make the information-bearing city name prosodically more salient and louder than the extraneous speech.

Table 1: End-pointing results on test-set utterances. For each utterance category, frequency of occurrence of each end-pointing result is shown.

	Utterance Category		
Result	Isolated city	Embedded city	Other speech
(1): % Missed	1.6	0.5	5.1
(2): % Divided	0.0	0.0	0.0
(3): % Clipped	0.3	0.9	6.5
(4): % Accurate	98.1	98.6	88.4

In addition to considering the behavior of this module where there is speech, it is relevant to consider what it does when there is no speech, as such *spurious regions* will be passed along to the recognizer just as the real words will. Despite the bias in favor of padding instead of clipping, and the presence of background and network noise on many utterances, the end-pointer still eliminated 81.1% of the non-speech utterances. In addition, it generated spurious regions in the non-speech portions of 0.8% of the utterances which contain speech. It should be recalled at this point that "non-speech" is at times a misnomer because background speech, televisions, radios etc. were not considered to be speech during the hand-labeling process.

Word Spotting

Our recognizer module is a word-spotting version of the Stochastic Explicit-Segment Modeling recognizer described by Leung [3]. It takes as input a speech region found by the end-pointer, and outputs whichever word in its vocabulary it finds best fits any portion of that region. Therefore, the recognizer module itself always provides some answer even for spurious regions and speech regions which do not contain city names in its vocabulary. It is the task of the rejection algorithm to discern that these outputs should be rejected, so the evaluation of the recognition module is only concerned with regions which contain words in the recognizer's vocabulary.

[1]We define an *utterance* to be the entire signal we receive from a user in response to our prompt, whether it contains speech or not.

For regions which contain in-vocabulary words not clipped by the end-pointer, the recognizer identified 88.8% (788/887) of isolated words[2] and 65.3% (96/147) of embedded words correctly; overall 85.5% of regions with in-vocabulary words were recognized correctly. Note that these statistics correspond most closely to recognition results traditionally reported in the literature, although those are usually based on laboratory speech rather than such real-user data as are used here.

Rejection

For each region processed by the recognizer, the rejection classifier uses a variety of the recognizer's intermediate parameters which may indicate the reliability of its final output, and decides whether the result should be rejected or accepted. The recognizer parameters are input to a mixture-Gaussian rejection classifier developed by Chigier [1]. The goal of the classifier is to reject regions for which the recognizer's output is wrong, which include: (1) spurious regions that passed through the end-pointer, (2) other regions lacking an in-vocabulary item, and (3) any vocabulary items mis-identified by the recognizer. The classifier computes a score representing a level of confidence in the recognizer's output, and makes its decision by comparing the score with a threshold. There is therefore a tradeoff between false acceptance and false rejection.

The cost of a false acceptance is high, because of likely customer dissatisfaction and additional operator time required to recover from such errors. For this reason, we use a high rejection threshold.

Rejection rates are shown by region type in Table 2. In summary, with our threshold setting, the rejection module rejects 92.5% of all regions which should indeed be rejected, at a cost of rejecting 36.1% of those that had in fact been recognized correctly. Alternatively viewed, the rejection module "filters" the recognizer outputs, taking as input regions of which 882/(656+882)=57.5% are correct, and outputting regions of which (882-318)/((882-318)+(656-607))=92.0% are correct.

Table 2: Rejection rates as function of recognition-module result and region type.

	Recognition Result	
Region Type	Incorrect	Correct
In-vocabulary isolated	90% (89/99)	34% (266/790)
In-vocabulary embedded	98% (49/50)	57% (52/92)
Out-of-vocab. and spurious	93% (469/507)	— —
Overall	92.5% (607/656)	36.1% (318/882)

4. SYSTEM EVALUATION

The purpose of the system evaluation is to characterize the behavior of the system as a function of the different categories of inputs it receives from the user. As shown in Figure 1, we consider four categories of user input, distinguishing whether the input contains target speech, non-target speech, both or neither. In this evaluation we distinguish three categories of output: rejection, correct acceptance, and false acceptance. Note that at the system level there is no difference between correct and false rejection. That distinction only exists within what we treat here as a "black box"; it is irrelevant to the application to which the system result is passed.

[2]Differences in experimental conditions as well as evaluation techniques contribute to differences between this result and those reported in the past.

Because utterances are sometimes segmented into multiple regions by the end-pointer, we evaluate the overall system in terms of utterances rather than regions. Thus, these results can be interpreted as statistics on the handling of users as a function of their degree of compliance[3] with the prompt for an isolated city name. Rejection of an utterance is defined to mean that either an utterance has no regions or else all of its regions are rejected. Correctness on an utterance is defined to be the accuracy on the first non-rejected region in that utterance. This approach is appropriate for a real-time system which acts on its first accepted input.

Results are summarized in Table 3. The low correct-acceptance rates are the combined consequence of user non-compliance and the (desirable) rejection of a large majority of inputs lacking a target word. As can be seen, correct acceptance occurs for 58.0% of the fully-compliant users. Adding successively less compliant users, correct acceptance occurs for (524+41)/(904+150)=565/1054=53.6% of all users who supply a city name in the recognizer's vocabulary, 565/(1054+446)=565/1500=37.7% of users who speak at all, or 565/(1500+401)=29.7% of all users.

Table 3: System evaluation results. For each utterance category, rates of rejection and correct and false acceptance are shown.

	Utterance Category; Number of Utterances			
System Results	In-Vocab. Isolated 904	In-Vocab. Embedded 150	Out of Vocabulary 446	No speech 401
Correct Acceptances	524 (58.0%)	41 (27.3%)	— —	— —
Rejections	370 (40.9%)	105 (70.0%)	412 (92.4%)	400 (99.8%)
False Acceptances	10 (1.1%)	4 (2.7%)	34 (7.6%)	1 (0.2%)

Focusing on accepted utterances, at a cost of not automating roughly 2/5 of the fully-compliant users, we can achieve 524/(524+10)=98.1% success on those compliant users whose requests are accepted by the system. Adding successively less compliant users, we find that (524+41)/(524+41+10+4)=97.6% of all accepted utterances with in-vocabulary city names are correct, 92.2% of accepted utterances with speech are correct, and 92.0% of all accepted utterances are correct.

5. INTERACTION BETWEEN WORD SPOTTING AND REJECTION

The module evaluations and system evaluation provide the most direct information about where system errors occur, by relating a single component's inputs to its outputs. Thus, we have answered questions such as "what fraction of target utterances does the end-pointer discard", "what fraction of correctly-recognized regions does the rejection algorithm reject", "what fraction of non-speech 'utterances' are eliminated by the combined effects of end-pointing and rejection", etc. The next level of detail is to look at instances in which one module performs imperfectly, not to the extent of making an erroneous decision such as mis-recognition, but rather to "weaken the status" of an utterance so as to increase the likelihood of error during subsequent processing. Thus, we address questions like "are clipped city names recognized less accurately than padded ones". In some cases, such as this one, a shortage of data in a needed category prevents analysis, as the end-pointer clipped very few city names here.

[3]Some "non-compliant" users by this definition are classified that way merely due to their city names not being in the top 25 rather than due to not providing a city name.

We address a different question of this type: "are correctly-recognized utterances more vulnerable to rejection when the word-spotter's hypothesized transcription misses part of the word?". The hypothesis being tested here is that the recognizer's parameters will reflect a poorer match between the signal and the recognizer's models in these cases, and so output appears to the rejection classifier to be more like a recognition error.

To address this issue, we extracted from the recognizer, in addition to the rejection algorithm's input parameters, the begin and end time of the hypothesized transcription corresponding to the recognizer's first-choice city name. We categorize these paths as either "clipped", meaning that at least 20 ms on one or both sides of the hand-labeled city name was omitted from the recognizer's transcription, or "correct". We limit attention to correctly-recognized city names which had not been clipped beforehand by the end-pointer.

The result is that clipping by the recognizer is common, and it appears to be associated with recognizer outputs which are more difficult for the rejection algorithm to distinguish from mis-recognized regions. Recognizer clipping occurred on 750 correctly-recognized regions; of these 278 (37.1%) were rejected. In contrast, only 28 (25.2%) of the 111 city names which were correctly delineated by the recognizer were rejected.

6. SUMMARY OF RESULTS

Through our evaluation of various components at various levels in our application, we have found the following results:

98.1% of accepted isolated-target-word utterances are correct,

92.0% of accepted utterances are correct,

88.8% of isolated-target-word regions are recognized correctly,

85.5% of target-word regions are recognized correctly,

58.0% of isolated-target-word utterances are recognized correctly and accepted,

53.6% of target-word utterances are recognized correctly and accepted,

37.7% of utterances with speech are recognized correctly and accepted, and

29.7% of utterances are recognized correctly and accepted.

As stated above, our strict rejection criterion provides high accuracy on accepted utterances (such as the 98.1% figure) at the cost of lower rates of acceptance (such as the 58.0% figure). User non-compliance is responsible for the discrepancy between overall and target-word correct-acceptance rates.

7. CONCLUSIONS

The most striking conclusion we draw is that the accuracy of a recognizer in a real-application scenario can be described in a wide variety of ways. Even without the additional complexity of evaluating a continuous-speech recognizer, we find that an assortment of statistics ranging from below 30% to over 98% can meaningfully describe various facets of "recognizer accuracy" for our particular system. We note further with concern that many of the descriptions of these statistics appear to be similar to each other, even when placed side-by-side.

As such, reporting "89% accuracy on 25 cities, real-user data" may be true, but it is misleading and far from an adequate description of our system's performance. The differences among many types of statistics seem subtle; consequently, a recognizer's performance can be portrayed virtually arbitrarily optimistically or pessimistically, by meaningful statistics. Therefore, comparison of recognition systems must be made more carefully, since comparisons of "bottom-line" accuracy statistics will be misleading when those statistics are not measured in exactly the same way, or when the statistics are not fully explained. Predicting application performance solely on the basis of fully-compliant and correctly end-pointed speech would unrealistically presume ideal behavior from (1) the other modules necessary for a real application, and (2) the users of that application. Furthermore, the high cost of false acceptance may require that rejection thresholds be set high, thus limiting acceptance rates but providing the requisite high accuracy on accepted utterances.

We conclude that it is important to choose the appropriate evaluation very carefully to fit the purpose of the evaluation. We have demonstrated module evaluations and a module interaction evaluation, which are appropriate for analyzing the accuracy of the recognizer and its peripheral algorithms. These evaluations must be focused on the appropriate classes of inputs to these modules — utterances, in the case of the end-pointing algorithm; regions containing in-vocabulary words, in the case of the recognizer; and all regions with accompanying recognizer output, in the case of the rejection module. We have described a system evaluation strategy to analyze the behavior of the overall system on the full range of inputs it will face in a real-application setting. In the future, system evaluation should be extended to *application evaluation*, taking into account issues such as the fact that Allston is part of Boston and so confusion between these, while a recognizer error, does not lead to an application error. Thus we will progress from questions like "what proportion of compliant users have their city names recognized correctly and accepted" to "what proportion of the traffic is correctly automated". Such an application evaluation can then be used as a basis for a more complete model of automation potential, encompassing such issues as system reliability and cost savings.

REFERENCES

1. Chigier, B., "Rejection and Keyword Spotting Algorithms for a Directory Assistance City Name Recognition Application", *Proceedings ICASSP 92: International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, California, March, 1992, v. II, pp. 93-96.

2. Hon, H.-W., *Vocabulary-Independent Speech Recognition: The VOCIND System*, Ph. D. Thesis, Carnegie-Mellon University, March, 1992, document # CMU-CS-92-108, see pp. 105-106.

3. Leung, H. C., I. L. Hetherington, and V. W. Zue, "Speech Recognition using Stochastic Explicit-Segment Modeling", *ICASSP 91: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, California, March, 1992, v. I, pp. 613-616.

4. Silverman, K. E. A., E. Blaauw, J. Spitz, and J. F. Pitrelli, "Towards Using Prosody in Speech Recognition/Understanding Systems: Differences Between Read and Spontaneous Speech", these proceedings.

5. Spitz, J., and the NYNEX Artificial Intelligence Speech Technology Group, "Collection and Analysis of Data from Real Users: Implications for Speech Recognition/Understanding Systems", *Proceedings of the 1991 DARPA Speech and Natural Language Workshop*, Pacific Grove, California, February, 1991, pp. 164-169.

6. Wilpon, J. G., L. G. Miller, and P. Modi, "Improvements and Applications for Key Word Recognition Using Hidden Markov Modeling Techniques", *Proceedings ICASSP 91: International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, Canada, May, 1991, v. 1, pp. 309-312.

ACKNOWLEDGEMENTS

Credit is due to much of the NYNEX Artificial Intelligence Laboratory's Speech Technology Group for assistance in preparing this presentation, including Ayman Asadi, Sara Basson, Ashok Kalyanswamy, Hong Leung, Kim Silverman, Judith Spitz, Steve Springer and Dina Yashchin.