

# International Journal of Speech Technology

## Predicting Search Term Reliability For Spoken Term Detection Systems

--Manuscript Draft--

<b>Manuscript Number:</b>	
<b>Full Title:</b>	Predicting Search Term Reliability For Spoken Term Detection Systems
<b>Article Type:</b>	Manuscript
<b>Keywords:</b>	spoken term detection, voice keyword search, information retrieval
<b>Corresponding Author:</b>	Joseph Picone, Ph.D. Temple University Elkins Park, Pennsylvania UNITED STATES
<b>Corresponding Author Secondary Information:</b>	
<b>Corresponding Author's Institution:</b>	Temple University
<b>Corresponding Author's Secondary Institution:</b>	
<b>First Author:</b>	Joseph Picone, Ph.D.
<b>First Author Secondary Information:</b>	
<b>Order of Authors:</b>	Joseph Picone, Ph.D. Amir Hossein Harati Nejad Torbati, MS
<b>Order of Authors Secondary Information:</b>	
<b>Abstract:</b>	Spoken term detection is an extension of text-based searching that allows users to type keywords and search audio files containing recordings of spoken language. Performance is dependent on many external factors such as the acoustic channel, language, pronunciation variations and acoustic confusability of the search term. Unlike text-based searches, the likelihoods of false alarms and misses for specific search terms, which we refer to as reliability, play a significant role in the overall perception of the usability of the system. In this paper, we present a system that predicts the reliability of a search term based on its inherent confusability. Our approach integrates predictors of the reliability that are based on both acoustic and phonetic features. These predictors are trained using an analysis of recognition errors produced from a state of the art spoken term detection system operating on the Fisher Corpus. We show that a 76% correlation between the predicted error rate and the actual measured error rate can be achieved, and that the remaining confusability is due to other acoustic modeling issues that cannot be derived from a search term's spelling.

## **PREDICTING SEARCH TERM RELIABILITY FOR SPOKEN TERM DETECTION SYSTEMS**

Amir Hossein Harati Nejad Torbati and Joseph Picone

*Abstract*— Spoken term detection is an extension of text-based searching that allows users to type keywords and search audio files containing recordings of spoken language. Performance is dependent on many external factors such as the acoustic channel, language, pronunciation variations and acoustic confusability of the search term. Unlike text-based searches, the likelihoods of false alarms and misses for specific search terms, which we refer to as reliability, play a significant role in the overall perception of the usability of the system. In this paper, we present a system that predicts the reliability of a search term based on its inherent confusability. Our approach integrates predictors of the reliability that are based on both acoustic and phonetic features. These predictors are trained using an analysis of recognition errors produced from a state of the art spoken term detection system operating on the Fisher Corpus. We show that a 76% correlation between the predicted error rate and the actual measured error rate can be achieved, and that the remaining confusability is due to other acoustic modeling issues that cannot be derived from a search term's spelling.

*Keywords*— spoken term detection, voice keyword search, information retrieval

Manuscript submitted December 30, 2012.

A. Harati and J. Picone are with the Department of Electrical and Computer Engineering at Temple University, 1947 North 12<sup>th</sup> Street, Philadelphia, Pennsylvania 19027 USA (phone: 215-204-4841; fax: 215-204-5960; email: joseph.picone@isip.piconepress.com).

## I. INTRODUCTION

The goal of a Spoken Term Detection (STD) system is “to rapidly detect the presence of a word or phrase in a large audio corpus of heterogeneous speech material” (Fiscus et al., 2007). As shown in Figure 1, STD systems typically index the audio data as a preprocessing step, allowing users to rapidly search the index files using common information retrieval approaches. Indexing can be done using a speech to text (STT) system (Miller et al., 2007), or simpler engines based on phoneme recognition (Nexidia, 2008). Like most detection tasks, STD can be characterized in terms of two kinds of errors: false alarms and missed detections (Martin et al., 1997). The overall error can be defined as a linear combination of these two errors. In this paper, we give equal weights to both types of errors.

Search engines have been used extensively to retrieve information from text files. Regular expressions (Duford, 1993) and statistically-based information retrieval algorithms (Manning et al., 2008) have been the foundations of such searches for many years. Text-based search algorithms use simple character recognition and character matching algorithms in which the identity of a character is known with probability 1 (no ambiguity). Unlike searching text data, searching through audio data requires handling ambiguity at the acoustic level. Determining the presence of a particular phone or word is not an exact science and must be observed through probabilities. A similarity measure used in such searches is typically based on some kind of score computed from a machine learning system. For text-based search systems, the performance of the system is independent of the term being searched (at least for a language like English where words are explicitly separated using spaces). For audio-based searches, however, the performance of the system depends on many external factors including the acoustic channel, speech rate, accent, language, vocabulary size and the inherent confusability of the search terms. Here we address the latter problem – predicting the reliability of a search term based on its inherent confusability.

The motivation for this work grew out of observations of typical users interacting with both word-based (Miller et al., 2007) and phone-based (Nexidia, 2008) voice keyword search systems over the past seven years. While it is well known that some aspects of search term performance, such as the duration of the word, correlate with search term performance (Doddington et al., 1999; Harati & Picone,

1  
2  
3  
4 2013), selecting robust and accurate search terms can be as much art as science. Users can quickly  
5 become frustrated because the nuances of the underlying speech processing engine don't always align  
6 with users' expectations based on their experiences with text-based searches. Therefore, our goal in this  
7 work was to develop a technology similar to password strength checking which displays the predicted  
8 strength of a keyword as a user types a search term. A demonstration of the system is available at  
9 [http://www.isip.piconepress.com/projects/ks\\_prediction/demo/current/](http://www.isip.piconepress.com/projects/ks_prediction/demo/current/). A screenshot of the user interface  
10 is shown in Figure 2.

11  
12  
13  
14  
15  
16  
17  
18  
19  
20 Our general approach in this work was to analyze error patterns produced by existing keyword search  
21 systems and to develop a predictive model of these errors. To build predictors of errors, we investigated  
22 both the acoustic phonetic distance between words and similarity measures of the underlying phone  
23 sequences. Our hypothesis for the acoustic phonetic approach was that acoustically similar words should  
24 have the same average error rate for a given speech recognizer. The similarity measure-based approach  
25 calculates an edit distance between the underlying phone sequences (Picone et al., 1990). These two  
26 approaches provided simple but useful baseline performance. A third approach, which is a major focus of  
27 this work, is based on extracting a variety of features from the spelling of a word and uses machine  
28 learning algorithms to estimate the error rate for that word.

29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40 A block diagram of our general approach is demonstrated in **Error! Reference source not found.**  
41 The input, a keyword search term that can consist of a word or phrase, is first transformed into features.  
42 These features result from the conversion of a word into several linguistic representations (e.g., phones,  
43 syllables). The preprocessor forms an augmented feature vector from an analysis of these linguistic  
44 representations (e.g., N-grams of phones or broad phonetic class). The machine learning block estimates  
45 one or more reliability scores, and passes these to the postprocessor for aggregation and normalization.  
46 For the machine learning task, we have implemented several statistical models based linear regression  
47 (Bishop, 2011), feed-forward neural networks (Bishop, 2011) and random forests (Breiman, 2001). The  
48 feature extraction process is central to this work since we have investigated what underlying linguistic  
49 properties of a word are the strongest predictors of search error rates. Since different approaches predict  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4 the error rate in different ways, we also explored combining predictors using a simple linear averaging  
5  
6 that employs particle swarm optimization (PSO) to find the optimal weights (Kennedy & Eberhart, 1995).  
7

## 8 9 **II. FEATURE GENERATION**

10 In this section we explore several approaches to generating features that can be used to measure the  
11  
12 similarity between words. Our goal is to determine feature combinations that have the highest correlation  
13  
14 with measured error rates.  
15

### 16 17 *A. Acoustic-Based Features*

18  
19 Based on our hypothesis that words with similar acoustic properties will result in similar error rates, one  
20  
21 possibility to predict the quality of a search term is to cluster words with similar acoustic properties and  
22  
23 average their associated error rates. We explored two ways to do this based on their acoustic and phonetic  
24  
25 properties. For an acoustic-based distance algorithm, the criterion used was a Euclidian distance in the  
26  
27 acoustic space. The acoustic space is constructed from features vectors based on a concatenation of  
28  
29 standard MFCC features (with derivatives and acceleration components) and duration (Young et al.,  
30  
31 2006; Davis & Mermelstein, 1980),  
32  
33

34  
35 The acoustic data was, of course, extracted from a different, non-overlapping corpus:  
36  
37 SWITCHBOARD (SWB) (Godfrey et al., 1992). A list of words was extracted from our target database,  
38  
39 the Fisher Corpus (Cieri et al., 2004). All instances of these words were located in SWB using the  
40  
41 provided time alignments (Deshmukh et al., 1998). Durations of the corresponding tokens were  
42  
43 normalized using a variation of an averaging approach developed by Karsmakers et al. (2007). Feature  
44  
45 vectors were constructed using three different approaches.  
46  
47

48  
49 In the first approach, each token was divided into three sections by taking its total duration in frames  
50  
51 and splitting that duration into three sections with durations arranged in 3-4-3 proportions (e.g., a token of  
52  
53 20 frames was split into three sections of lengths 6, 8 and 6 frames respectively). The average of the  
54  
55 corresponding feature vectors in each segment was computed, and the three resulting feature vectors were  
56  
57 concatenated into one composite vector. The final feature vector was obtained by adding the duration of  
58  
59  
60  
61  
62  
63  
64  
65

the token to the three 39-dimensional MFCC feature vectors, bringing the total dimension of the feature vector to  $3 \times 39 + 1 = 118$ .

We then created an alternate segmentation following the procedure described above that was based on a 10-24-32-24-10 proportion. This resulted in a feature vector of dimension  $5 \times 39 + 1 = 196$  elements. In our third approach, we divided the utterance into 10 equal-sized segments, which resulted in a feature vector of dimension  $39 \times 10 + 1 = 391$ .

Since there are so many word tokens, we used a combination of *K*-MEANS clustering and *k*-nearest neighbor classification (kNN) to produce an estimate of a test token's error rate. All feature vectors for a given word were clustered into *K* representative feature vectors, or cluster centroids, using *K*-MEANS clustering. We then used kNN classification to locate the *k* nearest clusters for a test token. The overall error rate for a word was computed as the weighted average of the *k* clusters, with the weighting based on an acoustic distance:

$$err(w_i) = A \hat{a}_{j \in D_k} \frac{1}{dist_{\text{Euclidean}}(w_i, w_j) + \epsilon} err(w_j), \quad (1)$$

$$A = \hat{a}_{j \in D_k} dist_{\text{Euclidean}}(w_i, w_j) + \epsilon, \quad (2)$$

where  $w_i$  is the word in question,  $D_k$  is the set of *k* nearest neighbors, and  $\epsilon$  is a small positive constant that guarantees the denominator will be non-zero.

### B. Linguistically-derived Features

A second approach based on a phonetic distance measure was also employed. Each token was converted into a phonetic representation using a dictionary or letter to sound rules (Elovitz et al., 1976). An edit distance (Wagner and Fischer, 1974) was computed using a standard dynamic programming approach. This approach was an attempt to model the underlying phonetic similarity between words, particularly compound words or words that shared morphemic representations.

Next we introduced a family of algorithms based on features extracted from the linguistic properties of words. These features included duration, length (number of letters), number of syllables, number of

1  
2  
3  
4 syllables/length, number of consonants/length, number of vowels/length, a ratio of the number of vowels  
5  
6 to the number of consonants, number of occurrences in the language model (count), monophone  
7  
8 frequency, broad phonetic class (BPC) frequency, consonant-vowel-consonant (CVC) frequency, biphone  
9  
10 frequency, 2-grams of the BPC and CVC frequencies, and 3-grams of the CVC frequencies. We have  
11  
12 used a simple phoneme-based duration model (Harati and Picone, 2013) to estimate the duration. The  
13  
14 correlation between duration and the average error rate is shown in Figure 4. The average error rate  
15  
16 decreases as the duration increases. However, as we will see shortly, the variance of this predictor is too  
17  
18 high to be useful in practical applications.  
19  
20

21  
22 The number of syllables was determined using a dictionary or syllabification software (Fisher, 1997)  
23  
24 for terms not in the dictionary. Mapping phones to consonant and vowel classes was easily accomplished  
25  
26 using a table lookup. The frequency of occurrence of a word, which we refer to as count, was measured  
27  
28 on the Fisher Corpus. A summary of the BPC classes used in our study is shown in Table 1. The  
29  
30 frequency measures used with these features consisted of the fraction of times each symbol appears in a  
31  
32 word. Next, we explore how these features were analyzed using several machine learning algorithms.  
33  
34

### 35 III. MACHINE LEARNING

36  
37 We evaluated three types of machine learning algorithms to map features to error rates. These algorithms  
38  
39 were chosen because they are representative of the types of learning algorithms available, provide a good  
40  
41 estimate of what type of performance is achievable, and also give us insight into the underlying  
42  
43 dependencies between features. Some have historical significance (e.g., linear regression) as a baseline  
44  
45 algorithm while others are known to provide state of the art performance (e.g., random forests).  
46  
47

48  
49 Linear regression (LR) (Bishop, 2011) is among the simplest methods that can be used to explore  
50  
51 dependencies amongst features. We assume that the predictive variable (e.g. error rate) can be expressed  
52  
53 as linear combination of the features:  
54

$$55 \quad y = Xb + e, \quad (3)$$

$$56 \quad \hat{b} = (X'X)^{-1} X'y. \quad (4)$$

1  
2  
3  
4 where  $X$  represents the input feature vector for a word,  $y$  represents the predicted error rate,  $e$  is the  
5 prediction error and  $b$  represents the weights to be learned from the training data.  
6  
7

8  
9 Feed-forward neural networks (NN) (Bishop, 2011) are among the most efficient ways to model a  
10 nonlinear relationship and have demonstrated robust performance across a wide range of tasks. As before,  
11 we assume a simple predictive relationship between  $X$  and  $y$ :  
12  
13

$$14 \quad y = f(X) + e . \quad (5)$$

15  
16 In our implementation,  $f()$ , the function to be estimated, is approximated as a weighed sum of sigmoid  
17 functions. We have used a network with one hidden layer. The output node is chosen to be linear.  
18  
19 Training was implemented the back-propagation algorithm.  
20  
21

22  
23 A random forest (RF) (Breiman, 2001) gives performance that is competitive with the best algorithms  
24 and yet does not require significant parameter tuning. The merits of the RF approach include speed,  
25 scalability and, most importantly, robustness to overfitting. A common approach for implementing a  
26 random forest is to grow many regression trees, each referred to as a base learner, using a probabilistic  
27 scheme. The training process for each base learner seeks the best predictor feature at each node from  
28 among a random subset of all features. A random subset of the training data is used that is constructed by  
29 sampling with replacement so that the size of the dataset is held constant. This randomization helps  
30 ensure the independence of the base learners. Each tree is grown to the largest extent possible without any  
31 pruning.  
32  
33

34  
35 RFs can also be used for feature selection using a bagging process that is implemented as follows. For  
36 one-third of trees in the forest, we generate the training subset using a special scheme: for the  $k^{\text{th}}$  tree we  
37 first put aside one-third of the data from the bootstrap process (sampling with replacement), and label this  
38 data out-of-bag (OOB) data. We apply the OOB data to each tree and compute the mean square error  
39 (MSE). Next, we randomly permute the value of a specific feature, rerun the OOB data, and compute the  
40 difference between old and new MSE. The value of this difference, averaged across all trees, shows the  
41 degree of sensitivity to this feature, and can be interpreted as the importance of that variable.  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65



#### IV. BASELINE EXPERIMENTS

The data used in this project was provided by BBN Technologies (BBN) and consisted of recognition output for the Fisher 2300-hour training set (Cieri et al., 2004). The speech recognizer was trained on 370 hours of SWB. The decoder used was configured to run 10 times faster than real time and was similar to a decoder used for keyword search (Miller et al., 2007). Recognition output consisted of word lattices, which we used to generate 1-best hypotheses and average duration information.

Though it is preferable to have disjoint training and evaluation sets, because the data available is limited, we used a cross-validation approach. We divided the data into 10 subsets and at each step use one of these subsets as the evaluation set and other 9 subsets as training data. At each step we trained models from a flat-start state using the corresponding training data. After rotating through all 10 subsets, we concatenated the results to obtain the overall estimate of performance. Statistics on both the training and evaluation sets are reported in terms of MSE, correlation and R values.

We have used two feature selection algorithms to explore which features are most important: sequential feature selection (the function `sequentialfs` in MATLAB) (Aha & Bankert, 1996) and random forests (the function `TreeBagger` in MATLAB) (Breiman, 2001). We began with a set of 150 features. We generated 7 subsets of these features as shown in Table 2. Set 1 was generated using sequential feature selection and linear regression with correlation as the criterion function. Set 2 was similar to set 1 except it used MSE as the criterion. Sets 3 and 4 used sequential feature selection with a neural network, with correlation and MSE as criteria. Sets 5 and 6 used a regression tree (built using the MATLAB function `RegressionTree.template`), with correlation and MSE as criteria respectively. Set 7 used the RF approach previously described. We see in Table 2 that approximately 50 features seems to be optimal but as few as 7 features gives reasonable performance. It also appears the training data is large enough to support these kinds of investigations as the results are well-behaved as a function of the number of features.

A plot of feature importance as determined by the RF algorithm is shown in Figure 5. Count, which represents the frequency of occurrence of a word, is recognized as the most important feature (its removal causes the highest increase in error.) Note that this does not mean that count is the most important feature

1  
2  
3  
4 in predicting the error rate. It simply means that the other features are highly correlated with each other,  
5  
6 so removing any one of these does not appreciably reduce the information content in the feature vector.  
7

8  
9 In Table 3, we present the correlation of the predicted error rates for the acoustic-based features using  
10 the *K*-MEANS/kNN approach previously described. In Table 4, we show results as a function of the  
11 number of nearest neighbors for the phonetic-based distance metric. Though the MSEs are comparable for  
12 both methods, the *R* values are higher for the acoustic-based metric, indicating a better prediction of the  
13 error rates. In Table 5, we compare three different classification algorithms as a function of the feature  
14 sets. The acoustic-based metric resulted in an *R* value of 0.6 on the evaluation set, while the  
15 phonetic-based methods resulted in an *R* value of 0.5, and the feature-based methods resulted in an *R* of  
16 0.7. The RF and NN classification methods resulted in similar *R* values.  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

## 27 V. SYSTEM COMBINATION

28 In order to investigate whether we can build a better predictor by combining different machines, we  
29 examined the correlation between predictors. As shown in Table 6, the acoustic-based distance is least  
30 correlated with the phonetic-based approach, indicating there could be a benefit to combining these  
31 predictors. We have explored combining systems using a weighted average of systems, where optimum  
32 weights are learned using particle swarm optimization (PSO) (Kennedy and Eberhart, 1995). The training  
33 process for PSO followed the same procedure described previously: the data, in this case word error rates  
34 for individual words, is divided into 10 equal subsets. One subset is used for evaluation, the remaining 9  
35 subsets are used for training, and the process is repeated by selecting each of the 10 subsets as the  
36 evaluation set. The 9 subsets are used to train 75 different classifiers representing a variety of systems  
37 selected across the three approaches (acoustic, phonetic and feature-based). PSO is applied to the  
38 predicted error rates produced by these 75 models on the held-out training data (referred to as  
39 development data). The result of this process is a vector representing the optimum weight of each  
40 machine. This process is repeated for each of the 10 partitions. The 10 vectors that result are then  
41 averaged together to produce the overall optimum weights. These weights are used to combine all 75  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4 machines into a single model. The error rate predictions of this model are then evaluated against the  
5  
6 reference error rates measured from the speech recognition output.  
7

8  
9 In this work we have a linearly constrained problem in which we want to find optimum weights for  
10  
11 our classifiers under the constraint that these weights sum to one. We have used Paquet and  
12  
13 Engelbrecht (2003) for this constrained optimization problem. In Table 7, we show the results obtained by  
14  
15 combining all 75 machines using PSO. These 75 machines are composed of 27 machines that use the  
16  
17 acoustic-based approach, 8 machines using the phonetic-based approach and 40 machines using the  
18  
19 feature-based approach. We also investigated removing the 8 linear regression machines, reducing the  
20  
21 number of systems from 75 to 67. This is shown in the second row of Table 7. The last three columns  
22  
23 show the percent that each machine contributes to the overall score.  
24  
25

26  
27 Acoustic-based and feature-based machines contribute equally to the overall score, and both  
28  
29 contribute significantly more than the phonetic-based approaches. In fact, when all 75 machines are  
30  
31 pooled, 43 of these machines (57%) have weights that are zero, implying they add no additional  
32  
33 information. The 43 machines included 12 from the acoustic-based machines (out of 27), 6 from the  
34  
35 phonetic-based machines (out of 8), and 25 from the feature-based machines (out of 40). By manually  
36  
37 excluding the 8 linear regression machines performance increases slightly. Prior to using PSO, our best  
38  
39 performance was an R value of 0.708. Our best R value with PSO and system combination was 0.761,  
40  
41 which is an improvement of 7.5%. Figure 6 shows the predicted error rate versus the reference error rate  
42  
43 for the system representing the second row of Table 7, demonstrating that there is good correlation  
44  
45 between the two.  
46  
47

## 48 49 VI. SUMMARY

50  
51 We have demonstrated an approach to predicting the quality of a search term in a spoken term  
52  
53 detection system that is based on modeling the underlying acoustic phonetic structure of the word. Several  
54  
55 similarity measures were explored (acoustic, phonetic and feature-based), as were several machine  
56  
57 learning algorithms (regression, neural networks and random forests). The acoustic-based and feature-  
58  
59 based representations gave relatively good performance, achieving a maximum R value of 0.7. By  
60  
61  
62  
63  
64  
65

1  
2  
3  
4 combining these systems using a weighted averaging process based on particle swarm optimization, the R  
5  
6 value was increased to 0.761.

7  
8 To further improve these results, we need to find better features. One of the more promising  
9  
10 approaches to feature generation involves an algorithm that predicts the underlying phonetic confusability  
11  
12 of a word based on inherent phone-to-phone confusions (Picone et al., 1990). We also, of course, need  
13  
14 more data, particularly data from a variety of keyword search engines. It is hoped that such data will  
15  
16 become available with the upcoming Spoken Term Detection evaluation to be conducted by NIST in  
17  
18  
19  
20 2013.

## 21 22 **VII. ACKNOWLEDGMENTS**

23  
24 The authors would like to thank Owen Kimball and his colleagues at BBN for providing the data  
25  
26 necessary to perform this study. This research was supported in part by the National Science Foundation  
27  
28 through Major Research Instrumentation Grant No. CNS-09-58854.

## 29 30 31 **VIII. REFERENCES**

- 32  
33 Aha, D. W., & Bankert, R. L. (1996). A comparative evaluation of sequential feature selection algorithms.  
34 In D. Fisher & H.-J. Lenz (Eds.), *Learning from Data: Artificial Intelligence and Statistics V* (1st  
35 ed., pp. 199–206). New York City, New York, USA: Springer.
- 36  
37 Bishop, C. (2011). *Pattern Recognition and Machine Learning* (2nd ed., p. 738). New York, New York,  
38 USA: Springer.
- 39  
40 Breiman, L. (2001). Random Forest. *Machine Learning*, 45(1), 5–32.
- 41  
42 Cieri, C., Miller, D., & Walker, K. (2004). The Fisher Corpus: a Resource for the Next Generations of  
43 Speech-to-Text. *Proceedings of the International Conference on Language Resources and*  
44 *Evaluation* (pp. 69–71). Lisbon, Portugal.
- 45  
46 Davis, S., & Mermelstein, P. (1980). Comparison of Parametric Representations for Monosyllabic Word  
47 Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and*  
48 *Signal Processing*, 28(4), 357–366.
- 49  
50 Deshmukh, N., Ganapathiraju, A., Gleeson, A., Hamaker, J., & Picone, J. (1998). Resegmentation of  
51 Switchboard. *Proceedings of the International Conference on Spoken Language Processing* (pp.  
52 1543–1546). Sydney, Australia.
- 53  
54 Doddington, G., Ganapathiraju, A., Picone, J., & Wu, Y. (1999). Adding Word Duration Information to  
55 Bigram Language Models. Presented at the IEEE Automatic Speech Recognition and Understanding  
56 Workshop. Keystone, Colorado, USA.
- 57  
58  
59  
60  
61  
62  
63  
64  
65

- 1  
2  
3  
4 Duford, D. (1993). *crep: a regular expression-matching textual corpus tool* (p. 84). Technical Report  
5 No. CUCS-005-93. Department of Computer Science, Columbia University, New York, New York,  
6 USA. doi: <http://hdl.handle.net/10022/AC:P:12304>.  
7
- 8 Elovitz, H., Johnson, R., McHugh, A., & Shore, J. (1976). *Automatic Translation of English Text to*  
9 *Phonetics by Means of Letter-to-Sound Rules* (NRL Report No. 7948) (p. 102). Washington, D.C.,  
10 USA. doi: <http://www.dtic.mil/dtic/tr/fulltext/u2/a021929.pdf>.  
11
- 12  
13 Fiscus, J., Ajot, J., Garofolo, J., & Doddington, G. (2007). Results of the 2006 Spoken Term Detection  
14 Evaluation. *Proceedings of the SIGIR 2007 Workshop: Searching Spontaneous Conversational*  
15 *Speech* (pp. 45–50). Amsterdam, Netherlands.  
16
- 17 Fisher, W. (1997). Tsylib syllabification package. url: <ftp://jaguar.ncsl.nist.gov/pub//tsylib2-1.1.tar.Z>. Last  
18 accessed on December 24, 2012.  
19
- 20  
21 Godfrey, J., Holliman, E., & McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for  
22 research and development. *Proceedings of the IEEE International Conference on Acoustics, Speech*  
23 *and Signal Processing* (pp. 517–520). San Francisco, California, USA.  
24
- 25 Harati, A., & Picone, J. (2013). Assessing Search Term Strength in Spoken Term Detection. To be  
26 presented at the IEEE International Multi-Disciplinary Conference on Cognitive Methods in  
27 Situation Awareness and Decision Support. San Diego, California, USA.  
28
- 29  
30 Karsmakers, P., Pelckmans, K., Suykens, J., & Van hamme, H. (2007). Fixed-Size Kernel Logistic  
31 Regression for Phoneme Classification. *Proceedings of INTERSPEECH* (pp. 78–81). Antwerp,  
32 Belgium.  
33
- 34 Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International*  
35 *Conference on Neural Networks* (pp. 1942–1948). Washington, D.C., USA.  
36
- 37 Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval* (p. 496).  
38 Cambridge, UK: Cambridge University Press.  
39
- 40  
41 Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). The DET curve in  
42 assessment of detection task performance. *Proceedings of Eurospeech* (pp. 1895–1898). Rhodes,  
43 Greece.  
44
- 45 Miller, D., Kleber, M., Kao, C.-L., Kimball, O., Colthurst, T., Lowe, S., & Schwartz, R. (2007). Rapid  
46 and Accurate Spoken Term Detection. *Proceedings of INTERSPEECH* (pp. 314–317). Antwerp,  
47 Belgium.  
48
- 49  
50 Nexidia, Inc. (2008). *Phonetic Search Technology* (p. 17). Atlanta, Georgia, USA. Retrieved from  
51 [http://www.nexidia.com/government/files/Static Page Files/White Paper Phonetic Search](http://www.nexidia.com/government/files/Static%20Page%20Files/White%20Paper%20Phonetic%20Search%20Tech%20.pdf)  
52 [Tech%20.pdf](http://www.nexidia.com/government/files/Static Page Files/White Paper Phonetic Search Tech%20.pdf).  
53
- 54 Paquet, U., & Engelbrecht, A. P. (2003). A new particle swarm optimiser for linearly constrained  
55 optimisation. *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 227–233).  
56 Canberra, Australia.  
57
- 58  
59 Picone, J., Doddington, G., & Pallett, D. (1990). Phone-mediated word alignment for speech recognition  
60 evaluation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(3), 559–562.  
61  
62  
63  
64  
65

1  
2  
3  
4 Wagner, R., & Fischer, M. J. (1974). The String-to-String correction problem. *Journal of the ACM*, 21(1),  
5 168–173.  
6

7  
8 Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., et al. (2006). *The HTK*  
9 *Book* (p. 384). Cambridge, U.K. (v3.4.1, url: <http://htk.eng.cam.ac.uk/docs/docs.shtml>).  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

## IX. LIST OF FIGURES

Figure 1. Spoken term detection can be partitioned into two tasks: indexing and search. One common approach to indexing is to use a speech to text system (after Fiscus et al., 2007).

Figure 2. A prototype of a web-based application that predicts voice keyword search term reliability is shown. The search term reliability is automatically updated as the user types a search term. A demonstration is available at [http://www.isip.piconepress.com/projects/ks\\_prediction/demo/current/](http://www.isip.piconepress.com/projects/ks_prediction/demo/current/).

Figure 3. In our approach to predicting search term reliability, we decompose terms into features, such as N-grams of phonemes and the number of phonemes, and apply these features to a variety of machine-learning algorithms.

Figure 4. The relationship between duration and error rate shows that longer words generally result in better performance, but the overall variance of this measure is high.

Figure 5. Feature importance based on the RF algorithm is shown. The feature "count," which represents the frequency of occurrence of a word, is by far the singlemost valuable feature since it is not correlated with any of the other features.

Figure 6. The predicted error rate is plotted against the reference error rate, demonstrating good correlation between the two.

## X. FIGURES

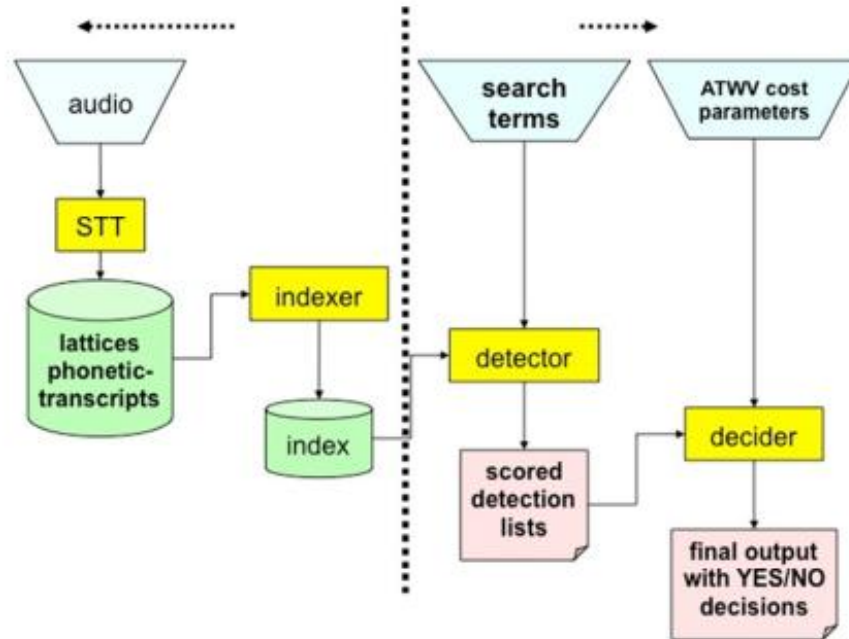


Figure 1. Spoken term detection can be partitioned into two tasks: indexing and search. One common approach to indexing is to use a speech to text system (after Fiscus et al., 2007).



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24




Keyword

Strength

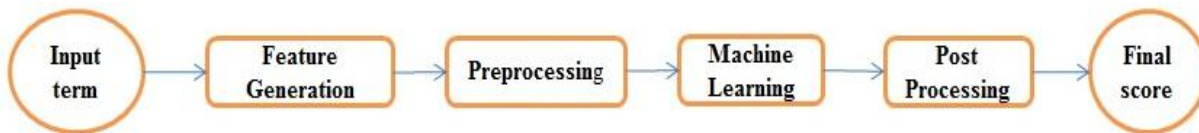
**Selecting a Good Voice Keyword Search Term**

- Longer words such as "brainstorm" are better than shorter words such as "brain."
- Words such as "onomatopoeia" that have more vowels and consonant-vowel transitions are better than words such as "turtle" that have fewer vowels.
- Frequently used words, such as "shopping" are better than less frequently used words such as "Chicago."
- Polysyllabic words such as "democracy" are better than monosyllabic words such as "vote."
- Multi-word phrases are better handled as structured queries using the intrinsic capabilities of your search tool.

**What is Voice Keyword Search(VKS)?**

- The voice signal is indexed using a pattern recognition algorithm such as a speech recognition system so that it can be quickly searched for content. 
- Accuracy varies based on a variety of linguistic and acoustic properties of the search term. 
- Voice queries are becoming increasingly popular for web-enabled phones and edge devices. 
- The demo is linked to the [Microsoft Research Audio Video Indexing System \(MAVIS\)](#).

25 Figure 2. A prototype of a web-based application that predicts voice keyword search term reliability is shown. The search term reliability is automatically updated as the user types a search term. A demonstration is available at [http://www.isip.piconepress.com/projects/ks\\_prediction/demo/current/](http://www.isip.piconepress.com/projects/ks_prediction/demo/current/).



36  
37  
38  
39  
40 Figure 3. In our approach to predicting search term reliability, we decompose terms into features, such as N-grams of phonemes and the number of phonemes, and apply these features to a variety of machine-learning algorithms.

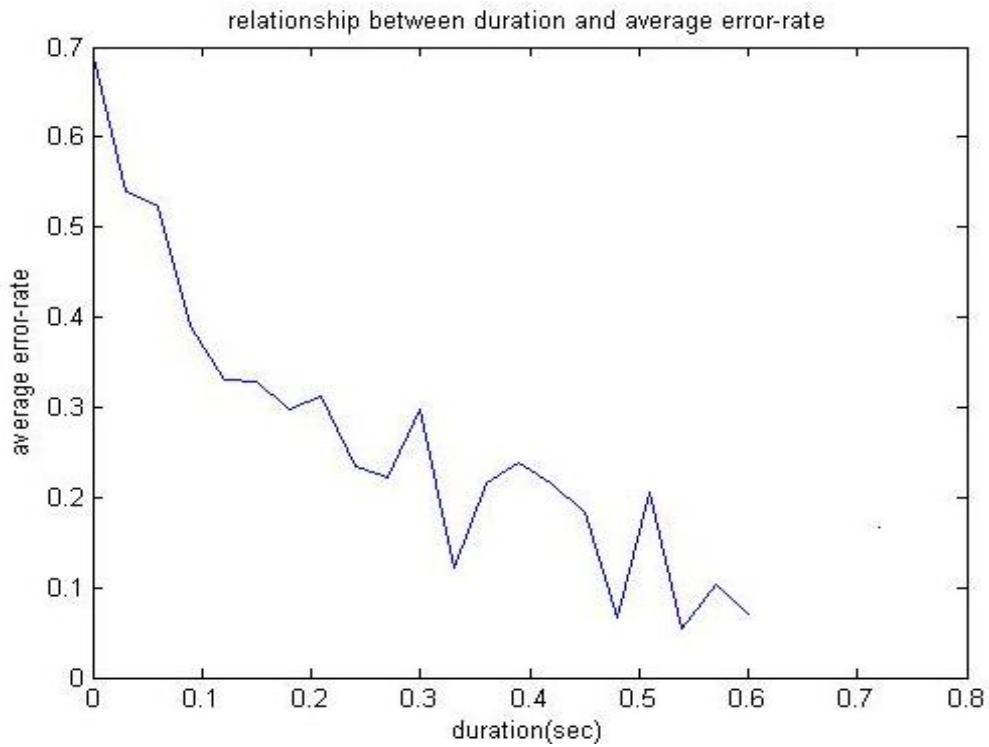


Figure 4. The relationship between duration and error rate shows that longer words generally result in better performance, but the overall variance of this measure is high.

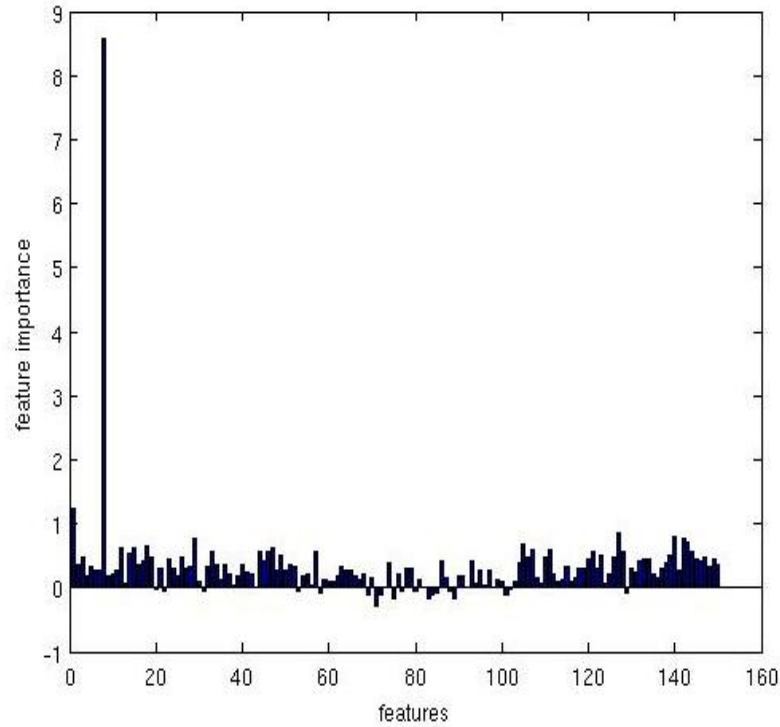


Figure 5. Feature importance based on the RF algorithm is shown. The feature "count," which represents the frequency of occurrence of a word, is by far the singlemost valuable feature since it is not correlated with any of the other features.

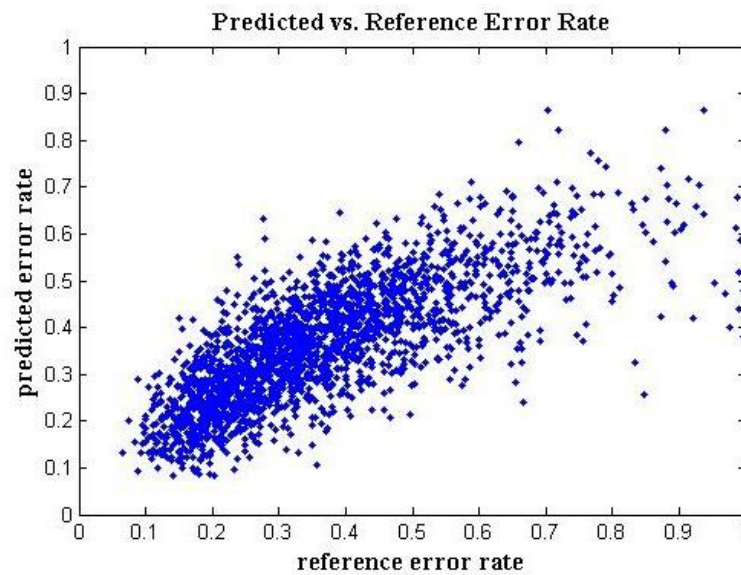


Figure 6. The predicted error rate is plotted against the reference error rate, demonstrating good correlation between the two.

## XI. LIST OF TABLES

Table 1. A mapping of phones to broad phonetic classes is shown.

Table 2. The number of features is shown for different feature selection methods as a function of the mean square error (MSE) on both the training and test sets. Performance for the correlation and MSE criteria was comparable.

Table 3. The correlation of predicted error rates with actual error rates is shown for our acoustic distance measure. Performance on the eval set is comparable for sets 1 and 2 for a broad range of parameter settings. The correlation between open set and closed set performance is also good.

Table 4. Results are shown for the phonetic distance algorithm as a function of the number of nearest neighbors used in kNN.

Table 5. A comparison of the different classification algorithms as a function of the feature sets is shown. R values are shown (the MSE results follow the same trend). Random forests (RF) give very stable results across a wide range of conditions.

Table 6. The correlation between various classifiers is shown. The acoustic-based distance is least correlated with the phonetic-based approach, indicating there could be a benefit to combining these predictors.

Table 7. Performance improves slightly by combining many predictors using PSO. The acoustic and feature-based metrics contribute equally to the overall result.

## XII. TABLES

Class	Phonemes
Silence	sp sil
Stops	b p d t g k
Fricatives	jh ch sh s z zh f th v dh hh
Nasals	m n ng en
Liquids	l e l r w y
Vowels	iy ih eh ey ae aa aw ay ah ao ax oy ow uh iw er

Table 2. A mapping of phones to broad phonetic classes is shown.

Method	No. Feats	MSE (Train)	MSE (Eval)
All Features / LR / Corr	150	0.015	0.018
SFS / LR / Corr	55	0.016	0.017
SFS / LR / MSE	54	0.016	0.017
SFS / NN / Corr	12	0.015	0.015
SFS / NN / MSE	14	0.015	0.015
SFS / Tree / Corr	7	0.015	0.020
SFS / Tree / MSE	7	0.016	0.019
RF	56	0.006	0.014

Table 1. The number of features is shown for different feature selection methods as a function of the mean square error (MSE) on both the training and test sets. Performance for the correlation and MSE criteria was comparable.

Set	K	k	Train		Eval	
			MSE	R	MSE	R
1	1	1	0.027	0.227	0.027	0.270
1	1	3	0.025	0.340	0.025	0.370
1	1	5	0.024	0.394	0.023	0.425
1	1	30	0.021	0.528	0.020	0.543
1	1	inf	0.023	0.456	0.022	0.471
1	2	1	0.026	0.293	0.025	0.330
1	2	3	0.024	0.414	0.023	0.444
1	2	5	0.022	0.461	0.022	0.473
1	2	30	0.019	0.569	0.019	0.583
1	2	inf	0.018	0.601	0.018	0.615
1	3	5	0.022	0.475	0.022	0.497
1	3	30	0.019	0.565	0.019	0.579
1	3	inf	0.018	0.600	0.018	0.614
1	4	5	0.022	0.477	0.021	0.499
1	4	30	0.020	0.542	0.020	0.559
1	4	inf	0.019	0.578	0.018	0.595
1	12	5	0.024	0.397	0.023	0.432
1	12	30	0.021	0.503	0.021	0.520
1	12	inf	0.021	0.519	0.020	0.542
2	2	5	0.024	0.387	0.024	0.407
2	4	inf	0.020	0.550	0.019	0.568
2	15	inf	0.021	0.526	0.020	0.551
2	17	inf	0.021	0.526	0.020	0.551

Table 3. The correlation of predicted error rates with actual error rates is shown for our acoustic distance measure. Performance on the eval set is comparable for sets 1 and 2 for a broad range of parameter settings. The correlation between open set and closed set performance is also good.

k	Train		Eval	
	MSE	R	MSE	R
1	0.026	0.296	0.026	0.322
3	0.024	0.405	0.024	0.421
5	0.023	0.434	0.023	0.451
30	0.021	0.502	0.021	0.519
50	0.021	0.503	0.021	0.519
100	0.021	0.499	0.021	0.515
300	0.022	0.483	0.022	0.498
inf	0.023	0.459	0.022	0.478

Table 4. Results are shown for the phonetic distance algorithm as a function of the number of nearest neighbors used in kNN.

Classifier	LR		NN		RF	
Method	Train	Eval	Train	Eval	Train	Eval
All Features / LR / Corr	0.683	0.618	0.724	0.624	0.895	0.708
SFS / LR / Corr	0.654	0.629	0.753	0.692	0.875	0.701
SFS / LR / MSE	0.654	0.629	0.735	0.686	0.857	0.697
SFS / NN / Corr	0.571	0.573	0.697	0.691	0.776	0.676
SFS / NN / MSE	0.573	0.574	0.697	0.689	0.799	0.679
SFS / Tree / Corr	0.561	0.564	0.674	0.669	0.761	0.659
SFS / Tree / MSE	0.561	0.564	0.674	0.669	0.761	0.659
RF	0.635	0.604	0.734	0.675	0.882	0.703

Table 5. A comparison of the different classification algorithms as a function of the feature sets is shown. R values are shown (the MSE results follow the same trend). Random forests (RF) give very stable results across a wide range of conditions.

	Acoustic	Phonetic	Feature
Acoustic	1	0.4	0.6
Phonetic	0.4	1	0.7
Feature	0.6	0.7	1

Table 7. The correlation between various classifiers is shown. The acoustic-based distance is least correlated with the phonetic-based approach, indicating there could be a benefit to combining these predictors.

Machines	Train		Eval		Relative Contribution		
	MSE	R	MSE	R	Acoustic	Phonetic	Feature
All	0.00092	0.913	0.012	0.760	41.1%	10.5%	48.3%
NN+RF	0.00084	0.918	0.012	0.762	44.7%	15.7%	39.5%

Table 6. Performance improves slightly by combining many predictors using PSO. The acoustic and feature-based metrics contribute equally to the overall result.

## LIST OF FIGURES

Figure 1. Spoken term detection can be partitioned into two tasks: indexing and search. One common approach to indexing is to use a speech to text system (after Fiscus et al., 2007).

Figure 2. A prototype of a web-based application that predicts voice keyword search term reliability is shown. The search term reliability is automatically updated as the user types a search term. A demonstration is available at [http://www.isip.piconepress.com/projects/ks\\_prediction/demo/current/](http://www.isip.piconepress.com/projects/ks_prediction/demo/current/).

Figure 3. In our approach to predicting search term reliability, we decompose terms into features, such as N-grams of phonemes and the number of phonemes, and apply these features to a variety of machine-learning algorithms.

Figure 4. The relationship between duration and error rate shows that longer words generally result in better performance, but the overall variance of this measure is high.

Figure 5. Feature importance based on the RF algorithm is shown. The feature "count," which represents the frequency of occurrence of a word, is by far the singlemost valuable feature since it is not correlated with any of the other features.

Figure 6. The predicted error rate is plotted against the reference error rate, demonstrating good correlation between the two.



## I. FIGURES

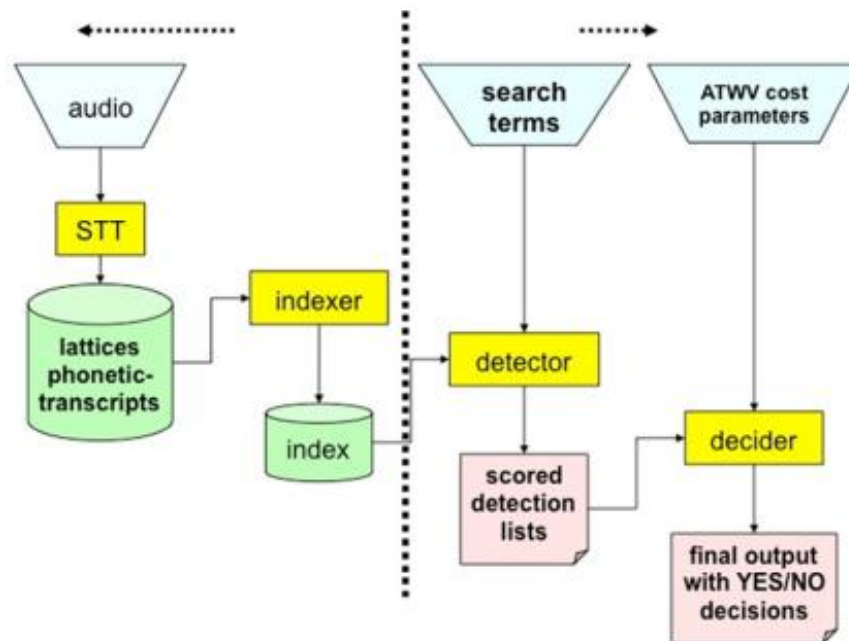


Figure 1. Spoken term detection can be partitioned into two tasks: indexing and search. One common approach to indexing is to use a speech to text system (after Fiscus et al., 2007).

The screenshot shows a web interface for predicting voice keyword search term reliability. On the left, there is a form with a 'Keyword' input field containing 'democrac', a 'Submit' button, and a 'Strength' progress bar showing 71% (7 out of 10 segments are green). Below the form is a section titled 'Selecting a Good Voice Keyword Search Term' with several bullet points:

- Longer words such as "brainstorm" are better than shorter words such as "brain."
- Words such as "onomatopoeia" that have more vowels and consonant-vowel transitions are better than words such as "turtle" that have fewer vowels.
- Frequently used words, such as "shopping" are better than less frequently used words such as "Chicago."
- Polysyllabic words such as "democracy" are better than monosyllabic words such as "vote."
- Multi-word phrases are better handled as structured queries using the intrinsic capabilities of your search tool.

On the right, a section titled 'What is Voice Keyword Search(VKS)?' contains three bullet points:

- The voice signal is indexed using a pattern recognition algorithm such as a speech recognition system so that it can be quickly searched for content. (Accompanied by a grid of small images)
- Accuracy varies based on a variety of linguistic and acoustic properties of the search term.
- Voice queries are becoming increasingly popular for web-enabled phones and edge devices. (Accompanied by an image of a hand holding a smartphone)

At the bottom right, there is a link: 'The demo is linked to the [Microsoft Research Audio Video Indexing System \(MAVIS\)](#).' (Accompanied by a logo with the text 'Search through the audio, not just text.' and a magnifying glass over a waveform).

Figure 2. A prototype of a web-based application that predicts voice keyword search term reliability is shown. The search term reliability is automatically updated as the user types a search term. A demonstration is available at [http://www.isip.piconepress.com/projects/ks\\_prediction/demo/current/](http://www.isip.piconepress.com/projects/ks_prediction/demo/current/).

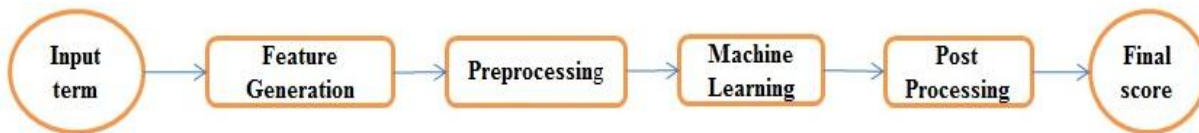


Figure 3. In our approach to predicting search term reliability, we decompose terms into features, such as N-grams of phonemes and the number of phonemes, and apply these features to a variety of machine-learning algorithms.

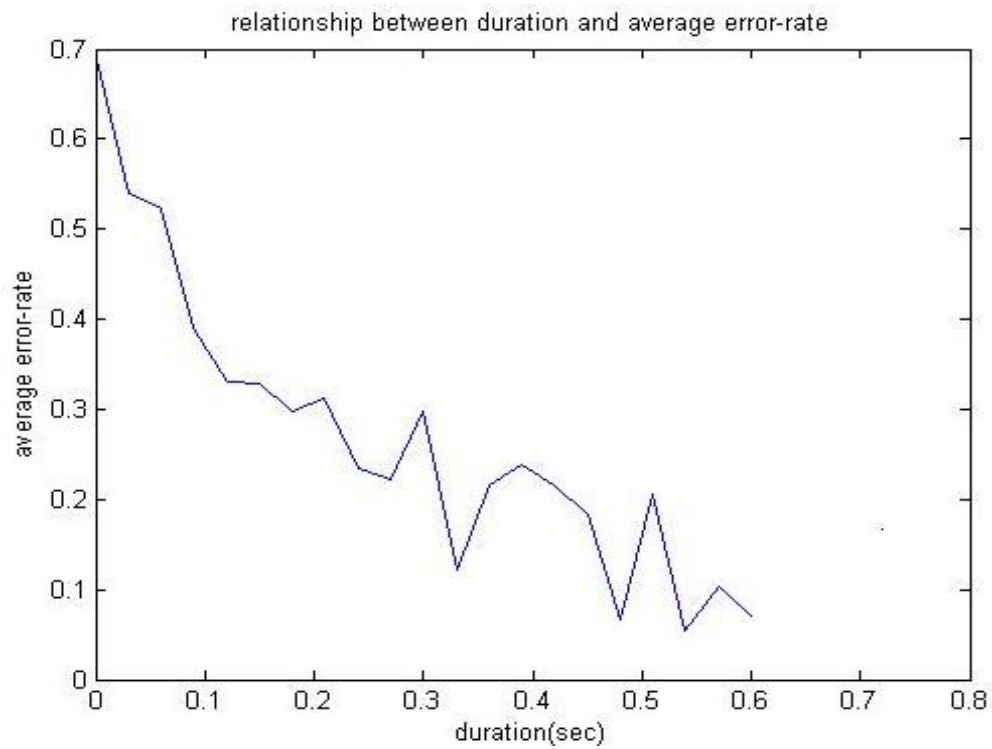


Figure 4. The relationship between duration and error rate shows that longer words generally result in better performance, but the overall variance of this measure is high.

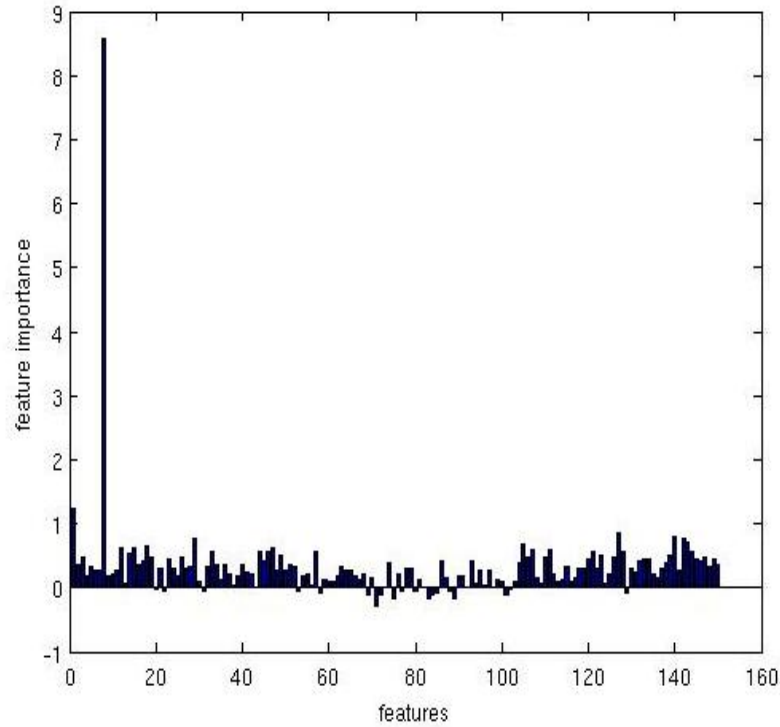


Figure 5. Feature importance based on the RF algorithm is shown. The feature "count," which represents the frequency of occurrence of a word, is by far the singlemost valuable feature since it is not correlated with any of the other features.

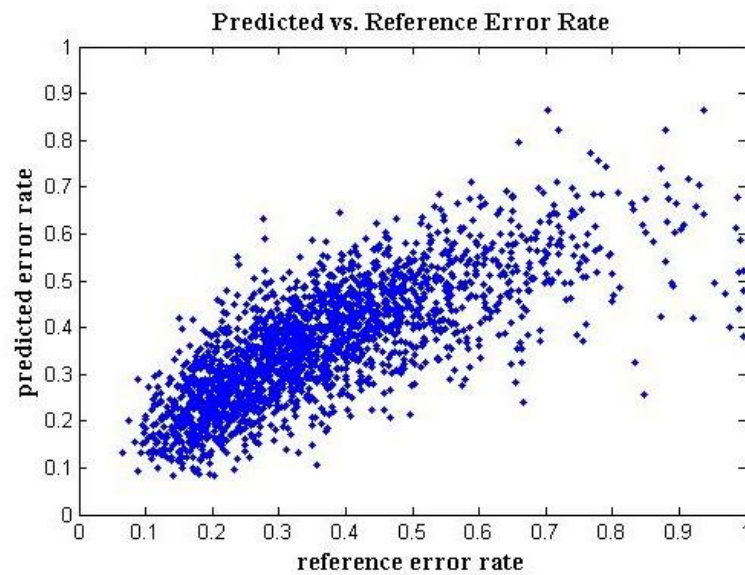


Figure 6. The predicted error rate is plotted against the reference error rate, demonstrating good correlation between the two.



## LIST OF TABLES

Table 1. A mapping of phones to broad phonetic classes is shown.

Table 2. The number of features is shown for different feature selection methods as a function of the mean square error (MSE) on both the training and test sets. Performance for the correlation and MSE criteria was comparable.

Table 3. The correlation of predicted error rates with actual error rates is shown for our acoustic distance measure. Performance on the eval set is comparable for sets 1 and 2 for a broad range of parameter settings. The correlation between open set and closed set performance is also good.

Table 4. Results are shown for the phonetic distance algorithm as a function of the number of nearest neighbors used in kNN.

Table 5. A comparison of the different classification algorithms as a function of the feature sets is shown. R values are shown (the MSE results follow the same trend). Random forests (RF) give very stable results across a wide range of conditions.

Table 6. The correlation between various classifiers is shown. The acoustic-based distance is least correlated with the phonetic-based approach, indicating there could be a benefit to combining these predictors.

Table 7. Performance improves slightly by combining many predictors using PSO. The acoustic and feature-based metrics contribute equally to the overall result.

## I. TABLES

Class	Phonemes
Silence	sp sil
Stops	b p d t g k
Fricatives	jh ch sh s z zh f th v dh hh
Nasals	m n ng en
Liquids	l e l r w y
Vowels	iy ih eh ey ae aa aw ay ah ao ax oy ow uh iw er

Table 2. A mapping of phones to broad phonetic classes is shown.

Method	No. Feats	MSE (Train)	MSE (Eval)
All Features / LR / Corr	150	0.015	0.018
SFS / LR / Corr	55	0.016	0.017
SFS / LR / MSE	54	0.016	0.017
SFS / NN / Corr	12	0.015	0.015
SFS / NN / MSE	14	0.015	0.015
SFS / Tree / Corr	7	0.015	0.020
SFS / Tree / MSE	7	0.016	0.019
RF	56	0.006	0.014

Table 1. The number of features is shown for different feature selection methods as a function of the mean square error (MSE) on both the training and test sets. Performance for the correlation and MSE criteria was comparable.

Set	K	k	Train		Eval	
			MSE	R	MSE	R
1	1	1	0.027	0.227	0.027	0.270
1	1	3	0.025	0.340	0.025	0.370
1	1	5	0.024	0.394	0.023	0.425
1	1	30	0.021	0.528	0.020	0.543
1	1	inf	0.023	0.456	0.022	0.471
1	2	1	0.026	0.293	0.025	0.330
1	2	3	0.024	0.414	0.023	0.444
1	2	5	0.022	0.461	0.022	0.473
1	2	30	0.019	0.569	0.019	0.583
1	2	inf	0.018	0.601	0.018	0.615
1	3	5	0.022	0.475	0.022	0.497
1	3	30	0.019	0.565	0.019	0.579
1	3	inf	0.018	0.600	0.018	0.614
1	4	5	0.022	0.477	0.021	0.499
1	4	30	0.020	0.542	0.020	0.559
1	4	inf	0.019	0.578	0.018	0.595
1	12	5	0.024	0.397	0.023	0.432
1	12	30	0.021	0.503	0.021	0.520
1	12	inf	0.021	0.519	0.020	0.542
2	2	5	0.024	0.387	0.024	0.407
2	4	inf	0.020	0.550	0.019	0.568
2	15	inf	0.021	0.526	0.020	0.551
2	17	inf	0.021	0.526	0.020	0.551

Table 3. The correlation of predicted error rates with actual error rates is shown for our acoustic distance measure. Performance on the eval set is comparable for sets 1 and 2 for a broad range of parameter settings. The correlation between open set and closed set performance is also good.

k	Train		Eval	
	MSE	R	MSE	R
1	0.026	0.296	0.026	0.322
3	0.024	0.405	0.024	0.421
5	0.023	0.434	0.023	0.451
30	0.021	0.502	0.021	0.519
50	0.021	0.503	0.021	0.519
100	0.021	0.499	0.021	0.515
300	0.022	0.483	0.022	0.498
inf	0.023	0.459	0.022	0.478

Table 4. Results are shown for the phonetic distance algorithm as a function of the number of nearest neighbors used in kNN.



Classifier	LR		NN		RF	
Method	Train	Eval	Train	Eval	Train	Eval
All Features / LR / Corr	0.683	0.618	0.724	0.624	0.895	0.708
SFS / LR / Corr	0.654	0.629	0.753	0.692	0.875	0.701
SFS / LR / MSE	0.654	0.629	0.735	0.686	0.857	0.697
SFS / NN / Corr	0.571	0.573	0.697	0.691	0.776	0.676
SFS / NN / MSE	0.573	0.574	0.697	0.689	0.799	0.679
SFS / Tree / Corr	0.561	0.564	0.674	0.669	0.761	0.659
SFS / Tree / MSE	0.561	0.564	0.674	0.669	0.761	0.659
RF	0.635	0.604	0.734	0.675	0.882	0.703

Table 5. A comparison of the different classification algorithms as a function of the feature sets is shown. R values are shown (the MSE results follow the same trend). Random forests (RF) give very stable results across a wide range of conditions.

	Acoustic	Phonetic	Feature
Acoustic	1	0.4	0.6
Phonetic	0.4	1	0.7
Feature	0.6	0.7	1

Table 7. The correlation between various classifiers is shown. The acoustic-based distance is least correlated with the phonetic-based approach, indicating there could be a benefit to combining these predictors.

Machines	Train		Eval		Relative Contribution		
	MSE	R	MSE	R	Acoustic	Phonetic	Feature
All	0.00092	0.913	0.012	0.760	41.1%	10.5%	48.3%
NN+RF	0.00084	0.918	0.012	0.762	44.7%	15.7%	39.5%

Table 6. Performance improves slightly by combining many predictors using PSO. The acoustic and feature-based metrics contribute equally to the overall result.