Author Query Form

Journal: IJST **Article ID:** IJST9197	**Please send your responses together with your list of corrections via web (preferred), or send the completed form and your marked proof to:** Akademijos 4, LT-08412 Vilnius, Lithuania fax: +370 5 2784 091 e-mail: vtexspr-corrections@vtex.lt

Dear Author,

During the preparation of your manuscript for typesetting, some questions have arisen. These are listed below.

Queries and/or remarks

Location in article (page/line)	Query / remark	Response
9/943	Please supply publisher for ref. Young et al. 2006	This "book" is really an online document. You could use Cambridge University as the publisher. However, it is really just a web resource.

Many thanks for your assistance

Metadata of the article that will be visualized in Online First

Please note: Images will appear in color online but will be printed in black and white.

Journal Name	International Journal of Speech Technology	
Article Title	Predicting search term reliability for spoken term detection systems	
Copyright holder	Springer Science+Business Media New York This will be the copyright line in the final PDF.	
Corresponding Author	Family name	Torbati
	Particle	
	Given Name	Amir
	Given Name	Hossein
	Given Name	Harati
	Given Name	Nejad
	Suffix	
	Division	Department of Electrical and Computer Engineering
	Organization	Temple University
	Address	1947 North 12th Street, Philadelphia, PA, 19027, USA
	E-mail	joseph.picone@isip.piconepress.com
Author	Family name	Picone
	Particle	
	Given Name	Joseph
	Suffix	
	Division	Department of Electrical and Computer Engineering
	Organization	Temple University
	Address	1947 North 12th Street, Philadelphia, PA, 19027, USA
	E-mail	
Schedule	Received	3 January 2013
	Revised	
	Accepted	17 May 2013
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. This work represents the first large-scale attempt to predict the success of a keyword search term from only its spelling. We explore the complex relationship between phonetic and acoustic properties of search terms. 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
Footnotes	

Int J Speech Technol
DOI 10.1007/s10772-013-9197-1

Predicting search term reliability for spoken term detection systems

Amir Hossein Harati Nejad Torbati · Joseph Picone

Received: 3 January 2013 / Accepted: 17 May 2013

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. This work represents the first large-scale attempt to predict the success of a keyword search term from only its spelling. We explore the complex relationship between phonetic and acoustic properties of search terms. 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

A.H.H.N. Torbati (✉) · J. Picone
Department of Electrical and Computer Engineering, Temple University, 1947 North 12th Street, Philadelphia, PA 19027, USA
e-mail: joseph.picone@isip.piconepress.com

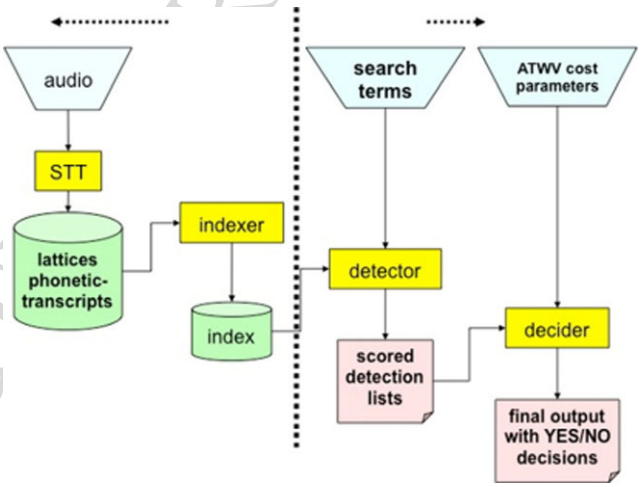

Fig. 1 Spoken term detection can be partitioned into two tasks: indexing and search. One common approach to indexing is to use a speech to text system (after Fiscus et al. 2007)

1 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 Fig. 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.

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 2/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

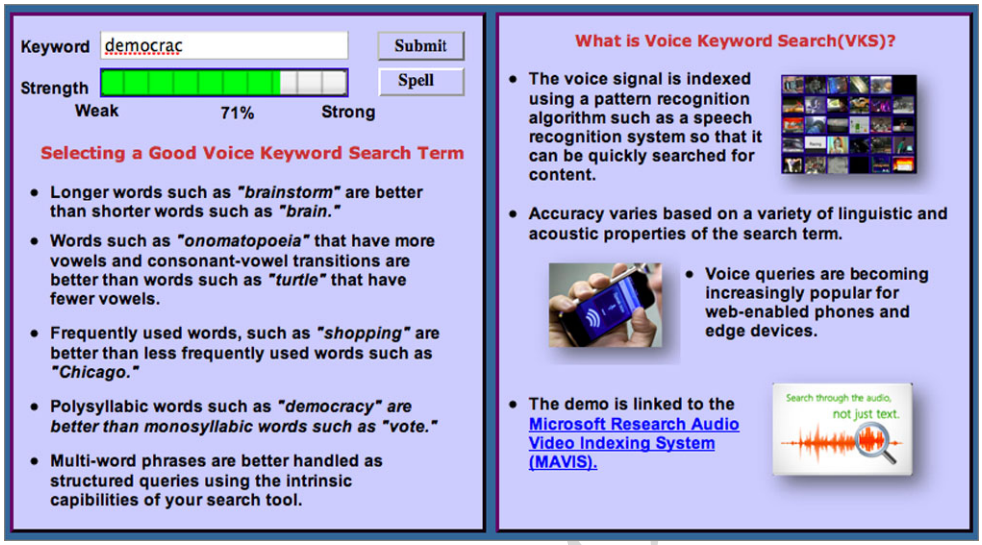

Fig. 2 A prototype of a web-based application that predicts voice keyword search term reliability is shown. The search term reliability is automatically updated as the user types a search term. A demonstration is available at http://www.isip.piconepress.com/projects/ks_prediction/demo/current/

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 only 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 wordbased (Miller et al. 2007) and phonebased (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 and Picone 2013), selecting robust and accurate search terms can be as much art as science. Users can quickly become frustrated because the nuances of the underlying speech processing engine don't always align with users' expectations based on their experiences with text-based searches. Therefore, our goal in this work was to develop a technology similar to password strength checking which displays the predicted strength of a keyword as a user types a search term.

A demonstration of the system is available at http://www.isip.piconepress.com/projects/ks_prediction/demo/current/. A screenshot of the user interface is shown in Fig. 2. The output of the tool is a visual feedback to the user in the form of a numeric score in the range [0,100 %] that indicates the quality of the search term (e.g., 100 % means the search term is strong and less likely to result in inaccurate hits). If a search term is likely to cause inaccurate results, that results in users having to sift through many utterances to find content of interest. The tool is an attempt to provide users with an interactive indication of the quality of a proposed term before they execute the search. Our experience with users is that, without this type of feedback, they often gravitate towards short search terms that are highly confusable. The tool makes it very easy for users to understand the value of selecting alternate search terms. Though not currently included in this tool, an obvious extension is to provide users with a list of alternate terms that are semantically similar yet have better reliability. Though we have not conducted extensive user evaluations with this tool, anecdotal results suggest that the feedback is very useful to casual users, and that users quickly understand the importance of selecting good search terms.

Our general approach in this work was to analyze error patterns produced by existing keyword search systems and to develop a predictive model of these errors. To build predictors of errors, we investigated both the acoustic phonetic distance between words and similarity measures of the underlying phone sequences. The use of acoustic measures resulted from a detailed analysis of the limited predictive power of phonetic or linguistic information. Our hypothesis for the acoustic phonetic approach was that acoustically

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 3/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

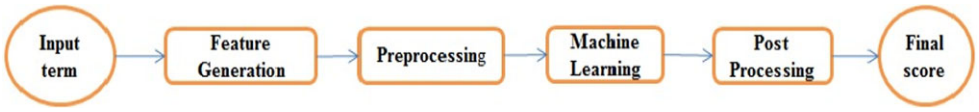

Fig. 3 In our approach to predicting search term reliability, we decompose terms into features, such as N-grams of phonemes and the number of phonemes, and apply these features to a variety of machine-learning algorithms

similar words should have the same average error rate for a given speech recognizer. The similarity measure-based approach calculates an edit distance between the underlying phone sequences (Picone et al. 1990). These two approaches provided simple but useful baseline performance. A third approach, which is a major focus of this work, is based on extracting a variety of features from the spelling of a word and uses machine learning algorithms to estimate the error rate for that word.

A block diagram of our general approach is demonstrated in Fig. 3. The input, a keyword search term that can consist of a word or phrase, is first transformed into features. These features result from the conversion of a word into several linguistic representations (e.g., phones, syllables). The preprocessor forms an augmented feature vector from an analysis of these linguistic representations (e.g., N-grams of phones or broad phonetic class). The machine learning block estimates one or more reliability scores, and passes these to the postprocessor for aggregation and normalization. For the machine learning task, we have implemented several statistical models based linear regression (Bishop 2011), feed-forward neural networks (Bishop 2011) and random forests (Breiman 2001). The feature extraction process is central to this work since we have investigated what underlying linguistic properties of a word are the strongest predictors of search error rates. Since different approaches predict the error rate in different ways, we also explored combining predictors using a simple linear averaging that employs particle swarm optimization (PSO) to find the optimal weights (Kennedy and Eberhart 1995).

The problem of predicting search term reliability is a relatively new problem and for the first time is addressed comprehensively in this paper. Researchers have often performed error analysis on speech recognition or keyword search experiments, but these have often been focused on system optimization and have been very specific to the data under consideration. The goal of the approaches explored in this paper is to develop a predictive tool that generalizes across corpora and can be used for vast audio archives found in YouTube and through search engines such as Google ad Bing. Hence, it is important that the methodology mix both linguistic and acoustic knowledge. In this paper, we present an extensive analysis of the predictive power of various types of features derived from this type of information.

2 Feature generation

In this section we explore several approaches to generating features that can be used to measure the similarity between words. Our goal is to determine feature combinations that have the highest correlation with measured error rates. Since this type of analysis is relatively new, there are no widely accepted set of baseline features for this problem. Our approach in this paper is to hypothesize a wide range of linguistic and acoustic features, and then to employ feature selection methods, discussed in Sect. 3, to select the most relevant ones.

2.1 Linguistically-derived features

Our original approach, motivated by the need to develop application-independent metrics, was based on a phonetic distance measure. 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 syllables/length, number of consonants/length, number of vowels/length, a ratio of the number of vowels to the number of consonants, number of occurrences in the language model (count), monophone frequency, broad phonetic class (BPC) frequency, consonant-vowel-consonant (CVC) frequency, biphone frequency, 2-grams of the BPC and CVC frequencies, and 3-grams of the CVC frequencies. We have used a simple phonemebased duration model (Harati and Picone 2013) to estimate the duration. The total number of linguistic features is 150, which includes a variety of N-grams of the above features.

The correlation between duration and the average error rate is shown in Fig. 4. The average error rate decreases as the duration increases. This correlates with our general experiences with users of these systems. On the surface, it would appear that the more syllables contained in a search term, the lesser its likelihood of being confused. However, as we will see shortly, the variance of this predictor is too

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 4/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

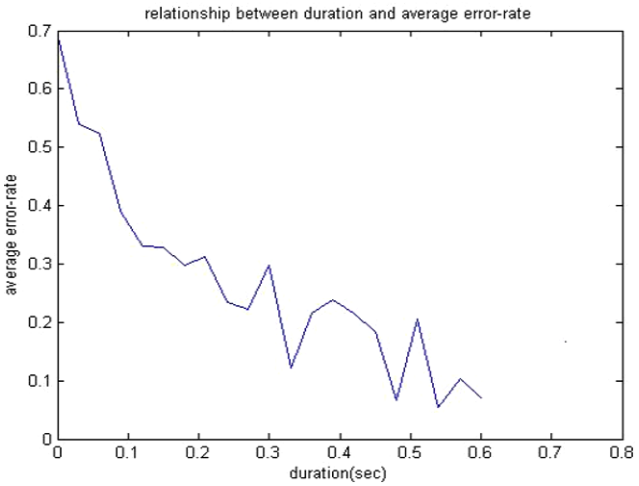

Fig. 4 The relationship between duration and error rate shows that longer words generally result in better performance, but the overall variance of this measure is high

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

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 el r w y
Vowels	iy ih eh ey ae aa aw ayah ao ax oy ow uh iw er

high to be useful in practical applications, due to some issues related to acoustic training in speech recognition.

The number of syllables was determined using a dictionary or syllabification software (Fisher 1997) for terms not in the dictionary. Mapping phones to consonant and vowel classes was easily accomplished using a table lookup. The frequency of occurrence of a word, which we refer to as count, was measured on the Fisher Corpus. A summary of the BPC classes used in our study is shown in Table 1. The frequency measures used with these features consisted of the fraction of times each symbol appears in a word.

2.2 Acoustic-based features

Based on our observation that linguistically-derived units had limited predictive power (to be explored more fully in Sect. 4), we hypothesized that words with similar acoustic properties will result in similar error rates. One possibility to exploit this behavior is to cluster words with similar acoustic properties and average their associated error rates. We explored two ways to do this based on their acoustic and phonetic properties. For an acoustic-based distance algorithm, the criterion used was a Euclidian distance in the acoustic space. The acoustic space is constructed from features vectors based on a concatenation of standard MFCC features (with derivatives and acceleration components) and duration (Young et al. 2006; Davis and Mermelstein 1980).

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

In the first approach, each token was divided into three sections by taking its total duration in frames and splitting that duration into three sections with durations arranged in 3–4–3 proportions (e.g., a token of 20 frames was split into three sections of lengths 6, 8 and 6 frames respectively). The average of the corresponding feature vectors in each segment was computed, and the three resulting feature vectors were concatenated into one composite vector. The final feature vector was obtained by adding the duration of the token to the three 39-dimensional MFCC feature vectors, bringing the total dimension of the feature vector to $3 * 39 + 1 = 118$.

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 $5 * 39 + 1 = 196$ elements. In our third approach, we divided the utterance into 10 equal-sized segments, which resulted in a feature vector of dimension $39 * 10 + 1 = 391$.

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:

$$err(w_i) = A \sum_{j \in D_k} \frac{1}{dist_{\text{Euclidean}}(w_i, w_j) + \varepsilon} err(w_j), \quad (1)$$

$$A = \sum_{j \in D_k} dist_{\text{Euclidean}}(w_i, w_j) + \varepsilon, \quad (2)$$

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

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 5/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

3 Machine learning

We evaluated three types of machine learning algorithms to map features to error rates. These algorithms were chosen because they are representative of the types of learning algorithms available, provide a good estimate of what type of performance is achievable, and also give us insight into the underlying dependencies between features. Some have historical significance (e.g., linear regression) as a baseline algorithm while others are known to provide state of the art performance (e.g., random forests). The models used in this paper can be regarded as a baseline for future research on this topic.

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

$$y = X\beta + \varepsilon, \tag{3}$$

$$\widehat{\beta} = \left(X'X\right)^{-1}X'y, \tag{4}$$

where X represents the input feature vector for a word, y represents the predicted error rate, ε is the prediction error and β represents the weights to be learned from the training data.

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

$$y = f(X) + \varepsilon. \tag{5}$$

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

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

RFs can also be used for feature selection using a bagging process that is implemented as follows. For one-third of trees in the forest, we generate the training subset using a special scheme: for the kth tree we first put aside one-third of the data from the bootstrap process (sampling with replacement), and label this data out-of-bag (OOB) data. We apply the OOB data to each tree and compute the mean square error (MSE). Next, we randomly permute the value of a specific feature, rerun the OOB data, and compute the difference between old and new MSE. The value of this difference, averaged across all trees, shows the degree of sensitivity to this feature, and can be interpreted as the importance of that variable.

4 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 and 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 (SFS) 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

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 6/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

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

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

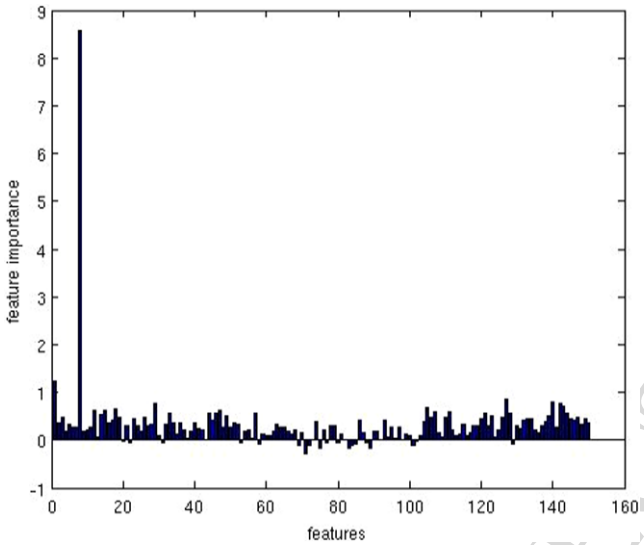

Fig. 5 Feature importance based on the RF algorithm is shown. The feature "count," which represents the frequency of occurrence of a word, is by far the single most valuable feature since it is not correlated with any of the other features

7 features gives reasonable performance. SFS selected features such as duration, length and count as the most relevant, particularly for the case of 7 features. 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 Fig. 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 relevant feature in predicting the error rate. It simply means that other features are highly correlated with each other, so removing any one of these does not appreciably reduce the information content in the feature vector. Figure 5 demonstrates that no individual feature stands out as having a large

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

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

predictive power. For example, N-grams of phonemes individually occur so infrequently that it is very hard for any one N-gram to influence the error rate. On the other hand, duration, length and other such aggregate features are correlated to each other and hence in combination don't provide a significant amount of new information. Therefore, we must explore more sophisticated combinations of these features.

In Table 3, we present the correlation of the predicted error rates for the acoustic-based features using the K-MEANS/kNN approach previously described. Performance is optimal for $K = 2$ and $k = \text{inf}$, which simply means the feature vectors were clustered into 2 clusters, and every element of each cluster was used in the kNN computation. However, overall performance is not extremely sensitive to the parameter settings, and the correlation of performance between the training and evaluation sets is good.

In Table 4, we show similar results as a function of the number of nearest neighbors for the phoneticbased distance metric. Though the MSEs are comparable for both methods, the R values are higher for the acoustic-based metric, indi-

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 7/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

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

Classifier method	No. feats	LR		NN		RF	
		Train	Eval	Train	Eval	Train	Eval
All Features/LR/Corr	150	0.683	0.618	0.724	0.624	0.895	0.708
SFS/LR/Corr	55	0.654	0.629	0.753	0.692	0.875	0.701
SFS/LR/MSE	54	0.654	0.629	0.735	0.686	0.857	0.697
SFS/NN/Corr	12	0.571	0.573	0.697	0.691	0.776	0.676
SFS/NN/MSE	14	0.573	0.574	0.697	0.689	0.799	0.679
SFS/Tree/Corr	7	0.561	0.564	0.674	0.669	0.761	0.659
SFS/Tree/MSE	7	0.561	0.564	0.674	0.669	0.761	0.659
RF	56	0.635	0.604	0.734	0.675	0.882	0.703

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

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

cating a better prediction of the error rates. This seems to indicate that acoustic modeling in speech recognition plays a more dominant role than the linguistic structure of a search term. Optimal performance is obtained with $k = 30$, which is on the order of the number of phonemes in our phoneme inventory, indicating that an excessive number of degrees of freedom are not needed in these feature sets.

In Table 5, we compare three different classification algorithms as a function of the feature sets. The acoustic-based metric resulted in an *R* value of 0.6 on the evaluation set, while the phonetic-based methods resulted in an *R* value of 0.5, and the feature-based methods resulted in an *R* of 0.7. The RF and NN classification methods resulted in similar *R* values. Approximately 80 % of the *R* value in these cases was due to duration. The remaining features accounted for a very small increase in the *R* value. There is no strong preference for features such as BPC and CVC since they were roughly comparable in their contribution to the overall *R* value.

The result of this section shows that some of the features like duration, count, bigram frequencies and acoustic distance have a relatively good correlation with the expected word error rate. A combination of these features can explain about 50 % of the variance in the prediction results. Our intuition indicates that duration reduces the acoustic ambiguity while bigram frequencies reflect both the occurrence of the word in the training database and the acoustic confusability of certain phoneme sequences.

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

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 %

5 System combination

In order to investigate whether we can build a better predictor by combining different machines, we examined the correlation between predictors. As shown in Table 6, the acoustic-based distance is least correlated with the phonetic-based approach, indicating there could be a benefit to combining these predictors. We have explored combining systems using a weighted average of systems, where optimum weights are learned using particle swarm optimization (PSO) (Kennedy and Eberhart 1995). The training process for PSO followed the same procedure described previously: the data, in this case word error rates for individual words, is divided into 10 equal subsets. One subset is used for evaluation, the remaining 9 subsets are used for training, and the process is repeated by selecting each of the 10 subsets as the evaluation set. The 9 subsets are used to train 75 different classifiers representing a variety of systems selected across the three approaches (acoustic, phonetic and feature-based). PSO is applied to the predicted error rates produced by these 75 models on the held-out training data (referred to as development data). The result of this process is a vector representing the optimum weight of each machine. This process is repeated for each of the 10 partitions. The 10 vectors that result are then averaged together to produce the overall optimum weights. These weights are used to combine all

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 8/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

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

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

75 machines into a single model. The error rate predictions of this model are then evaluated against the reference error rates measured from the speech recognition output.

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

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

6 Summary

We have demonstrated an approach to predicting the quality of a search term in a spoken term detection system that is based on modeling the underlying acoustic phonetic structure of the word. Several similarity measures were explored (acoustic, phonetic and feature-based), as were several machine learning algorithms (regression, neural networks and random forests). The acoustic-based and feature-based representations gave relatively good performance, achieving a maximum R value of 0.7. By combining these systems using a weighted averaging process based on particle swarm optimization, the R value was increased to 0.761.

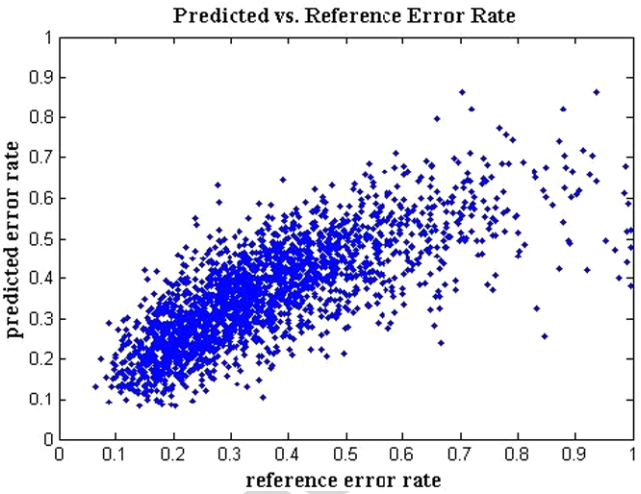

Fig. 6 The predicted error rate is plotted against the reference error rate, demonstrating good correlation between the two

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

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

References

Aha, D. W., & Bankert, R. L. (1996). A comparative evaluation of sequential feature selection algorithms. In D. Fisher & H.-J. Lenz (Eds.), *Learning from data: artificial intelligence and statistics V* (1st edn., pp. 199–206). New York: Springer.

Bishop, C. (2011). *Pattern recognition and machine learning* (2nd edn., p. 738). New York: Springer.

Breiman, L. (2001). Random forest. *Machine Learning*, *45*(1), 5–32.

Cieri, C., Miller, D., & Walker, K. (2004). The fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings of the international conference on language resources and evaluation*, Lisbon, Portugal (pp. 69–71).

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357–366.

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF

« IJST 10772 layout: Large v.1.3.2 file: ijst9197.tex (Andrius) class: spr-twocol-v1.4 v.2013/04/17 Prn:2013/05/29; 12:58 p. 9/9»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: apa»

Deshmukh, N., Ganapathiraju, A., Gleeson, A., Hamaker, J., & Picone, J. (1998). Resegmentation of switchboard. In *Proceedings of the international conference on spoken language processing*, Sydney, Australia (pp. 1543–1546).

Doddington, G., Ganapathiraju, A., Picone, J., & Wu, Y. (1999). Adding word duration information to bigram language models. Presented at the IEEE automatic speech recognition and understanding workshop, Keystone, Colorado, USA.

Duford, D. (1993). *Crep: a regular expression-matching textual corpus tool* (Technical Report No. CUCS-005-93) (p. 84). Department of Computer Science, Columbia University, New York, USA. http://hdl.handle.net/10022/AC:P:12304.

Elovitz, H., Johnson, R., McHugh, A., & Shore, J. (1976). *Automatic translation of English text to phonetics by means of letter-to-sound rules* (NRL Report No. 7948) (p. 102). Washington, DC, USA. http://www.dtic.mil/dtic/tr/fulltext/u2/a021929.pdf.

Fiscus, J., Ajot, J., Garofolo, J., & Doddingtion, G. (2007). Results of the 2006 spoken term detection evaluation. In *Proceedings of the SIGIR 2007 workshop: searching spontaneous conversational speech*, Amsterdam, Netherlands (pp. 45–50).

Fisher, W. (1997). Tsylb syllabification package. url: ftp://jaguar.ncsl.nist.gov/pub//tsylb2-1.1.tar.Z. Last accessed on 24 December 2012.

Godfrey, J., Holliman, E., & McDaniel, J. (1992). SWITCHBOARD: telephone speech corpus for research and development. In *Proceedings of the IEEE international conference on acoustics, speech and signal processing*, San Francisco, California, USA (pp. 517–520).

Harati, A., & Picone, J. (2013). Assessing search term strength in spoken term detection. To be presented at the IEEE international multi-disciplinary conference on cognitive methods in situation awareness and decision support, San Diego, California, USA.

Karsmakers, P., Pelckmans, K., Suykens, J., & Van hamme, H. (2007). Fixed-size kernel logistic regression for phoneme classification. In *Proceedings of INTERSPEECH*, Antwerp, Belgium (pp. 78–81).

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, Washington, DC, USA (pp. 1942–1948).

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (p. 496). Cambridge: Cambridge University Press.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of Eurospeech*, Rhodes, Greece (pp. 1895–1898).

Miller, D., Kleber, M., Kao, C.-L., Kimball, O., Colthurst, T., Lowe, S., & Schwartz, R. (2007). Rapid and accurate spoken term detection. In *Proceedings of INTERSPEECH*, Antwerp, Belgium (pp. 314–317).

Nexidia, I. (2008). *Phonetic search technology* (p. 17). Atlanta, Georgia, USA. Retrieved from http://www.nexidia.com/government/files/StaticPageFiles/WhitePaperPhoneticSearchTech%2Epdf.

Paquet, U., & Engelbrecht, A. P. (2003). A new particle swarm optimiser for linearly constrained optimisation. In *Proceedings of the IEEE congress on evolutionary computation*, Canberra, Australia (pp. 227–233).

Picone, J., Doddington, G., & Pallett, D. (1990). Phone-mediated word alignment for speech recognition evaluation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *38*(3), 559–562.

Wagner, R., & Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM*, *21*(1), 168–173.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., et al. (2006). In *The HTK book* (p. 384). Cambridge: ???. v3.4.1, url. http://htk.eng.cam.ac.uk/docs/docs.shtml.

Journal ID: 10772, Article ID: 9197, Date: 2013-05-29, Proof No: 1, UNCORRECTED PROOF