Applying the Naïve Bayes Classifier to Assist Users in Detecting Speech Recognition Errors

Lina Zhou  
Information Systems Department  
UMBC  
Baltimore, MD 21250  
zhoul@umbc.edu

Jinjuan Feng  
Information Systems Department  
UMBC  
Baltimore, MD 21250  
jfeng2@umbc.edu

Andrew Sears  
Information Systems Department  
UMBC  
Baltimore, MD 21250  
asears@umbc.edu

Yongmei Shi  
Computer Science and Electrical Engineering Department  
UMBC  
Baltimore, MD 21250  
yshi1@umbc.edu

Abstract
Speech recognition (SR) is a technology that can improve accessibility to computer systems for people with physical disabilities or situation-introduced disabilities. The wide adoption of SR technology; however, is hampered by the difficulty in correcting system errors. HCI researchers have attempted to improve the error correction process by employing multi-modal or speech-based interfaces. There is limited success in applying raw confidence scores (indicators of system’s confidence in an output) to facilitate anchor specification in the navigation process. This paper applies a machine learning technique, in particular Naïve Bayes classifier, to assist detecting dictation errors. In order to improve the generalizability of the classifiers, input features were obtained from generic SR output. Evaluation on speech corpuses showed that the performance of Naïve Bayes classifier was better than using raw confidence scores.

Keywords  
Speech recognition, disability, error identification, Naïve Bayes classifier, assistive technology

1. Introduction
Speech-based technology allows hand-free human computer interaction that is not possible when using the standard keyboard and mouse. This is particularly significant to users with limited hand functions. For example, many diseases or injuries may cause the partial or complete loss of hand functions but do not affect speech, such as spinal cord injury, missing limbs, and repetitive strain injury. Moreover, speech-based interaction increases accessibility for people in circumstances when the use of their hands is hindered [1], such as driving a vehicle. SR technologies have already been applied to assist children with learning disabilities and older people with reduced motor skills [2-6]. Thus, speech-based solutions become critical approaches to interacting with computer and even to conducting daily activities. The effectiveness of speech-based solutions relies on the advancement of speech recognition (SR) technologies.

The focus of this study is continuous dictation recognition systems. The context represents a challenging speech recognition task, which involves large vocabulary and an unlimited number of context domains. SR technologies for dictation are still far from perfect and errors in recognition must be expected. Recent studies suggest that recognition accuracy continues to be problematic when using state-of-the-art SR technology under realistic conditions [7]. The consistent presence of SR errors makes it necessary to take an extra step to find and correct recognition errors.

Manual identification and correction of errors is very time-consuming [7, 8]. The correction process is extremely frustrating [9], especially to students with learning disabilities. It becomes more problematic for users who depend solely on speech-based interactions and have little access to complementary communication modalities. It often happens that the person who generates speech input is not the one who proofreads the transcriptions (e.g., witnesses’ depositions in a lawyer’s office are transcribed by office staff instead of the witnesses themselves). Even more challenges are present for manual error correction when original audio files are not available and the transcription must strictly follow the exact words that were spoken. Therefore, we need to look
for alternative solutions to assist users in correcting errors in SR transcription.

Some researchers in the HCI (human-computer interaction) community have made efforts to improve navigation in the error correction process. They often provide navigation environments empowered by speech-based solutions [10] or multi-modal solutions [11]. One navigation solution employs the concept of navigation anchors with simple commands including move right, move down, and next. These approaches are promising but all suffer from one limitation: they require users to identify errors. A navigation-based solution [12] took a step toward associating navigation anchors, which are used to pre-specify default cursor positions, with possible errors identified using raw confidence scores. The approach received limited success but highlighted the need for automated techniques to improve the accuracy of error detection in SR output. They have yet to employ the potential of machine learning techniques in the automatic prediction of errors.

To address the limitations of the pure HCI approach, we employ a machine learning technique, Naïve Bayes classifier, to automatically detect recognition errors. We selected Naïve Bayes not only because it is one of the most efficient and effective inductive learning algorithm for classification [13], but also because the performance of overall classification output is likely to be dominated by the class with lower population [14]. In our context, errors are our primary concern, which constitute smaller population. We do not attempt to completely automate the recognition correction process but aim to facilitate user-based error identification and correction activities.

We evaluated the performance of Naïve Bayes technique on two speech corporuses collected from prior studies using a commercial SR system. The results are very encouraging. On average, the machine learning model can identify over 74 percent of the errors, and 43.5 percent of the identified errors were truly errors. There is strong practical significance in the research result. First and foremost, the intelligent error identification model can be incorporated into navigation systems to support users in error navigation and correction. Furthermore, the Naïve Bayes classifier can be integrated into SR engines to improve the performance of the underlying SR technologies.

The rest of the paper is organized as follows. In Section 2, we give a brief overview of the importance of SR technologies to people with disabilities. We also discuss the challenges of error detection in SR and previous research in this area. In Section 3, configurations and input features of the Naïve Bayes classifier are presented in detail. In Section 4, we evaluate the classifier on two speech corpora. Finally, we highlight insights gained from this study and suggest directions for further enhancement.

2. Background

Speech can be used in both text generation and command and control, which are probably two most common activities when people interact with computer. Generating text from the speech input is mostly used for composition purpose. A typical example of the former application is dictating business letters. In contrast, command and control issues command for computer systems to take actions. There are a wide range of applications of such kind of systems including cursor control, menu control, appliance control, and so on. Cursor control should control the location and focus of the cursor on the screen [10]. Appliance control controls the operations household appliance such as microwaves [15].

2.1 Importance of speech recognition to people with disabilities

Even though computers have been widely used and become an inseparable part of our daily life, they are not applicable to everyone under all conditions. People with physical disabilities (e.g. spinal cord injuries) may have difficulties in using traditional input devices, such as the key board and the mouse, to access the computer. A recent report on statistics of specific disabilities in the United States [16] showed that the number of people with disabilities affecting the computer usage is about five time of that of the people with visual disabilities, whom are often considered as people with most urgent need to access the computer, and this number steadily increased in recent years. Even people with no physical disabilities also experience difficulties in using traditional devices when their hands are busy or when the environments interfere with the use of hand functions.

Hand-free voice interface is an obvious and promising alternative to the traditional input devices for those people. Automatic SR systems are needed to process the input speech signals and convert them into text. While the reverse process, Text-to-Speech, has been broadly used to aid people with visual disabilities to use computers by synthesizing speech from the text shown on the screen. Evidence has been provided to demonstrate the positive impact of speech recognition on the accessibility of disabled people. Drawing on the extant literature, we summarized the application of SR technology to improve the accessibility of disabled people in Table 1.

<table>
<thead>
<tr>
<th>Application</th>
<th>Disability type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice navigation</td>
<td>General</td>
<td>SAP Workplace [16], Controlled web browsing [17]</td>
</tr>
<tr>
<td>Speech understanding</td>
<td>Motor control</td>
<td>SUITEkeys [18]</td>
</tr>
</tbody>
</table>
Table 1. Application of SR to disabled people

<table>
<thead>
<tr>
<th>Interface</th>
<th>Condition</th>
<th>Project/Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR dictation</td>
<td>Spinal Cord Injuries</td>
<td>Sears et al [7]</td>
</tr>
<tr>
<td>Word processing</td>
<td>Rheumatoid Arthritis</td>
<td>A TIPP II Project [3]</td>
</tr>
<tr>
<td>Hand-free typing</td>
<td>Repetitive Stress Injury</td>
<td>VocalProgramming [20]</td>
</tr>
</tbody>
</table>

The SAP Workplace [16] is a voice interface that allows voice navigation in a complex business GUI. Several techniques were used to prevent and resolve the ambiguities caused by same name and label for multiple entries in complex GUI, as well as balance the user efficiency. Christian et al [17] explored the possibility of voice controlled web browsing. Subjects read the anchor test or the number associated with a link to browse the corresponding web page. The results showed that voice controlled web browsing had the low error rate, but took 1.5 times longer than mouse controlled browsing.

SUITEdkeys [18] is a speech understanding interface for motor-control, which challenged users to fully accessing the functionalities of the keyboard and the mouse. In the system, a natural language grammar was constructed to model the keyboard/mouse objects and their actions. The speech inputs accepted by the grammar correspond to keyboard/mouse actions. The pilot study with three persons with upper-body motor-control impairments showed that even with minimal training, participants got good performance using speech input.

Sears et al [7] compared the effectiveness of SC in dictation task when used by people with Spinal Cord Injuries (SCI) with people without disability. The participants with SCI in the study had no or very limited motor functions in their arms and hands. All the participants finished two types of tasks: transcriptions and compositions. The results showed that participants with SCI gave more positive response with respect to the time and effort required to finish both tasks. They also showed more positive attitude toward the overall ease of use of the system and the SR method as compare with their normal method to interact with computer.

Learning is a lifetime endeavor for people with or without disabilities. It is particularly challenging for students with learning disabilities to keep pace with other students in regular classroom settings. To address this problem, Liberated Learning Project [19] incorporated the SR technology into university classroom. The spoken lecture could be simultaneously processed by SR system and converted into text that were displayed for students’ reference. Through this process, SR also provided an alternative to traditional note taking for students with disability.

A TIPP II Project [3] have been using SR to help students who are among the most severely learning disabled in Ontario of Canada. There were several successful use of SR system in the school [9]. For example, one student used SR in order to write independently at an adequate speed. Another student, who had limited hand function because of Rheumatoid Arthritis that caused pain and fatigue in her hands after working on homework for 20-30 minutes, used SR for word-processing in preparation for a greater workload in high school. A third student had letter recognition and word retrieval difficulties, and found reading and writing to be extremely laborious and frustrating. He used an SR system to help him write faster and better without having to worry about spelling words correctly. A SR writing workshop [2] was constructed to include students with learning disability who have writing problems, and normal achieving students. The comparison between the two groups of students showed that students with learning disability as a whole improved in writing fluency and thought they could write better when using SR.

Repetitive Stress Injury (RSI) affects people whose jobs involve long period of typing, such as computer programmers. VocalProgramming [20] is a system that allows the programmers to do the programming by voice. A context-free grammar was built for the programming language, and a limited programming language vocabulary is used to enhance the performance of the speech recognition.

There are many other situations that introduce temporary disabilities to ordinary people (e.g., when a doctor’s hands are busy with operating on a patient), where SR may play a significant role. However, the potential of SR technologies have not been fully realized mainly due to technical challenges, in particular, the existence of recognition errors [12]. As a result, people have to go through error correction process to improve SR output, which could be both time-consuming [8] and frustrating [9].

2.2 Challenges of Error Detection

Error correction is a complex process, which can be divided into three consecutive phases [10]: users must detect that a recognition error has occurred, then navigate to the error, and finally correct the error. If the noise introduced in one phase is not resolved, it cascades to the following phases. Thus, error detection is crucial to the result of error correction. We focus on error detection in this paper.

System accuracy is a major challenge facing SR users, especially people with disabilities. It is shown that for students without disability, even if SR error occurs, they can validate it using the audio cues. But for the students
with a hearing disability, the SR errors may cause the misunderstanding. Experiments showed that in order to generate an accurate document, the time spent on editing was three times of that spent on original audio length [19]. Therefore, high SR accuracy is vital to truly improve the accessibility of disabled people. Since errors still permeate in the output of commercial large-vocabulary SR dictation systems, error correction becomes a necessary following-up step to remedy the imperfectness of SR output.

Most successful commercialized speech applications focus on small vocabularies or limited domains. In comparison, error detection in large vocabulary dictation SR system is more challenging for the following reasons:

- It is continuous and involved with a large vocabulary. Dictation speech shows no clear boundary between sentences and phrases. Punctuation, an important indicator of sentence delimiters, is likely to be ignored in this kind of applications. Misrecognition of sentence boundary could lead to misunderstanding of the entire sentence. Disfluency insertion and ambiguity in interpreting audio signals may further confuse SR system. Moreover, the presence of large vocabulary implies a large number of possibilities in translating a speech into words, which reduces the probability of choosing the right output.

- Local context may not be useful in signifying errors. A major reason is that errors can occur in long sequences as a result of commonly used tri-gram HMM decoder models. In other words, the surrounding words of an error may still be errors. In some cases, a word could be completely wrong semantically but still grammatically sound.

- Some information in the original speech input, which is helpful to detect errors, may not be available during error detection process.

The current research attempts to answer the question: what techniques can we use to help users in detecting errors?

2.3 Related Work

Researchers from multiple disciplines (e.g., HCI, speech processing, language models, and text mining) have attempted to improve error detection process.

The HCI community focuses on designing systems to help SR users in improving the efficiency and effectiveness of interactive error correction process. To address the limitation of speech-only interaction, multimodal interface has been designed to facilitate correcting errors in speech user interface [11]. New speech-based approach was also designed to help users in navigating to recognition errors [12], which explored the potential of using confidence scores to assign navigation anchors and simplify navigation.

Confidence scores are used to provide feedback to recognition algorithms. Confidence scores indicate SR system’s confidence about an output. Since a sequence of acoustic signals can potentially be transformed into various combinations of words, the system can only select the best one as the output. Confidence scores are produced by extracting confidence features from the computation of word and utterance hypotheses, which are then utilized by a classifier to accept/reject the recognition hypotheses. Confidence scores have been defined at different levels (e.g., word level [21], sentence level [22], often using probabilistic approaches [23] that incorporate phonetic or prosodic information and acoustic features [24]. Theoretically speaking, the higher the confidence score, the greater the output. However, this is not always the case. Our statistical evaluation of SR output shows that one eighth of system output does not have the highest confidence scores [12]. This exposes an inherent limitation of raw confidence score-based approach. In addition, the performance of such an approach highly depends on the selection of cut-off thresholds, which tend to vary across users and systems. For example, it was found [10] that the accuracy in classifying a word into recognition error or correct recognition was maximized at the threshold of –5, which resulted in only 12% of the recognition errors being identified and 55% of identified errors were truly errors. 69% of the recognition errors can be identified with the threshold of 3, but the percentage of true errors dropped to 36%. The results were encouraging but not satisfactory.

A significant amount of work in speech processing has focused on error reduction in SR output [25, 26]. For example, 53% of unrecognized and out-of-vocabulary words were detected in a 1000 utterance test set, while rejecting only 6% of the correct words, by using acoustic probabilities [27]. The error detection rate was improved to 59% after applying semantic, pragmatic, and discourse level information. They employed a small lexicon (1,800 words), which simplifies the process of identifying out-of-vocabulary errors and correcting errors. This does not meet the needs of real-world dictation. Moreover, they employed all kinds of knowledge resources, including acoustic features, phonetic features, lexical features, grammatical features, domain-specific features (e.g., flight names in air traffic systems), and so on, in reducing errors. Some of the knowledge sources are not available to users who are identifying and correcting errors. They also made it impossible to look for solutions without delving into the details of the back-end SR engines.

We propose a “black-box” approach, which not only takes advantage of the generic information from SR output but also addresses the limitations of raw confidence score-based approach. The idea is to use machine learning techniques to post-process the hypotheses generated by SR systems to identify errors. It is common for SR systems to generate alternative
hypotheses and their confidence scores in addition to the final output. By utilizing such information, we can develop classifiers to automatically learn whether the best hypothesis generated by a SR engine is to be accepted or rejected. In view of the wide availability of such information, the proposed approach can be extended to other SR systems. Therefore, our research question becomes: based on the generic information in SR output, can a machine learning technique produce better performance than a raw confidence-based approach in error detection?

3. Naïve Bayes Classifiers

We experimented with Naïve Bayes (NB) classifier to predict errors in the SR output. The technique is selected for three reasons:

1) NB favors the minority population of the data [14], which happen to be the focus of this study. If we group SR output into two categories: correct recognition and errors, errors that we are interested in detecting account for a small proportion of the output.

2) NB has been applied by a prior study in reducing SR errors [28].

3) The NB classifier is one of the most efficient and effective inductive learning algorithm for classification [13].

3.1 Design of Naïve Bayes Classifier

A Bayesian network is a directed, acrylic graph that compactly represents a probabilistic distribution [29]. A directed edge between two nodes indicates a probabilistic influence (dependency) from the variable denoted by the parent node to that of the child. Consequently, the structure of the network denotes the assumption that each node $X_i$ in the network is conditionally independent of its non-descendants given its parents.

A Bayesian classifier is simply a Bayesian network applied to a classification task. It contains a node $C$ representing the class variable and a node $X_i$ for each feature. Given a specific instance $x$ (an assignment of values $x_1, x_2, \ldots, x_n$ to the feature variables), the Bayesian network allows us to compute the probability $P(C=c_k | X=x)$ for each possible class $c_k$, $k=0$ (correct recognitions) or 1 (errors). This is done via Bayes theorem, giving us:

$$P(C=c_k | X=x) = \frac{P(X=x | C=c_k)P(C=c_k)}{P(X=x)} \tag{1}$$

A critical component in Formula (1), $P(X=x | C=c_k)$, is impractical to compute without imposing restrictive assumptions. A Naïve Bayesian classifier [30] applies an independence assumption, which assumes that each feature $X_i$ is conditionally independent of every other feature, given the class variable $C$. Formally, this yields:

$$P(X=x | C=c_k) = \prod_{i=1}^{n} P(X_i=x_i | C=c_k) \tag{2}$$

There has been a great deal of work on developing methods for learning networks specifically for classification tasks (e.g., [31, 32]). These approaches allow for a limited form of dependence between feature variables, so as to relax the restrictive assumptions of the Naïve Bayesian classifier. In this paper, we focus on using the Naïve Bayesian classifier, but present possibility for learning richer probabilistic classification models in future work.

3.2 Feature Selection

We first select various features from the SR output so as to arrive at better classification models. These features can be grouped into four categories: confidence scores, alternative hypotheses, context information, and domain features. All of them are generic, can be directly derived from the SR output.

Confidence scores are unique for each output word and for each alternative hypothesis. So are confidence scores for each utterance, which consists of a sequence of words. However, further decisions need to be made on the scope of the other three types of information. In order to get a better idea of the significance of the recognition alternatives to error detection, we first investigated the chance that an alternative instead of final output word is correct. Figure 1 shows the relationship between the coverage of correct recognition and the first $n$ ($0 \leq n \leq 10$) candidates (SR system output corresponds to $n=0$).

![Figure 1. The relationship between the coverage of correct recognition and the first n output candidates](image)

It is shown that, on average, 82.9% of the best alternatives are the words that were actually spoken. Approximately 13.3% of the time, the correct word does not exist in the first 10 alternatives produced by the speech recognition system. The curve in Figure 1 flattens out after the first three alternatives. Thus, we incorporate three best alternative candidates and use the difference in
their confidence scores between two consecutive hypotheses as input features.

When it comes to contextual information, we apply a heuristic approach. Since traditional tri-gram models employed by SR engines consider 3-word context, we explored with a 7-word window involving 3 immediate predecessor words and 3 immediate successor words of the current word. Contextual information was represented with differential confidence scores, which are defined as the differences in confidence score between the current word and another word in the context. For example, the following word of the current word was incorporated as the differential confidence scores between the following word and the current word.

Since we are dealing with general-purposed large-vocabulary SR systems, domain-specific knowledge does not apply well. Nonetheless, the distribution of confidence scores was employed to reflect the overall level of system’s confidence in an output.

NB is known for assuming independence between input features. Little dependency was expected to exist between the selected input features. For example, the differential confidence score between an output word and its first alternative hypothesis has nothing to do with the differential confidence score between the first alternative hypothesis and the second hypothesis and the between the output word and the preceding word. Even if there is some kind of dependency, regardless of its strength, it was found that NB can still be optimal if the dependencies distribute evenly in classes, or if the dependences cancel each other out [13].

Feature selection is an important step in data mining. An ideal feature space is both sufficient and necessary. During feature selection, features that do not contribute to improving classification performance or lead to reducing classification performance should be removed. Feature selection helps to attenuate the degree to which the assumption of learning algorithms is violated (e.g., independence assumption in the Naïve Bayesian classifier [33]).

In view of the binary classification of system output, we applied logistic regression model [34] to estimate importance weights for each input features. As a result, we selected word confidence scores, utterance confidence scores, differential confidence scores of the first two alternative hypotheses, and differential confidence scores of two words preceding the current word and one word after the current word as input features to train the NB classifier.

4. Evaluations

4.1 Experimental Setup

We tested the efficacy of the error detection models with 5M corpus collected from two previous studies on spontaneous speech dictation task [35]. DICT-I corpus consists of documents from 15 participants’ real dictation on general topics using IBM’s ViaVoice speech recognition engine (Millennium Edition), recorded under high-quality conditions. The documents contain a total of 66950 words, including 55495 words that were correctly recognized and 11455 recognition errors. DICT-II corpus was generated by 12 participants working under similar conditions; however, participants were not allowed to make inline correction. In other words, they can only correct errors after the dictation was completed. These documents contain a total of 4865 words, including 761 recognition errors.

The participants in both studies were all native speakers. None were professional announcers or reporters. To ensure objectivity, misrecognized words (errors) were first marked by the participants themselves and then validated by researchers via cross-referencing the recorded speech.

In the documents, each word is assigned an integer confidence score. In addition, alternatives were generated that may be correct if the “most likely” word was incorrect. We recorded both the “most likely” word and the “10-best alternatives” along with their confidence scores. The first alternative generated is considered the “best alternative.”

Each data set was split into a training set and an evaluation set, where the data from two-thirds of participants were used for training and the remaining one-third of participants for evaluation or testing. In other words, the speakers for the training set were different from those for the testing set.

4.2 Test Methods

Given the training data and the initial feature sets, we conducted feature selection for Naïve Bayes (NB) classifiers. Since we were focusing on error detection, the following evaluation metrics were used to evaluate the performance of NB classifiers: precision (PRE) and recall (REC). PRE is the percentage of words classified as errors that are in fact recognition errors. REC denotes the proportion of actual recognition errors in the test set that are categorized as error by the classifier. Values of the above two measures tend to move in opposite directions that improvement on one measure usually leads to degradation on the other. F-measure (F-M), a single measure combining precision and recall, can be used as an efficient single-valued metric. F-M is commonly used in information retrieval [36], which is defined as follows:

\[
F - M = \frac{2 \times PRE \times REC}{PRE + REC}
\]

Two types of errors could occur in predicting recognition errors. One is false acceptance that corresponds to words that are correctly decoded in the
recognizer but accepted (as errors) by the classifiers, and the other is false rejection corresponding to incorrectly decoded words that are rejected (as correct words) by the classifiers. An ideal classifier would verify all the correct results while rejecting all false alarms. However, there is always a trade-off between the two types of errors.

4.3 Results

Table 2 reports the performance of NB classifiers on two speech corpora. The best error detection results are listed in terms of PRE, REC, and F-M.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Recall</th>
<th>Precision</th>
<th>F-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>DICT-I</td>
<td>0.75</td>
<td>0.42</td>
<td>0.53</td>
</tr>
<tr>
<td>DICT-II</td>
<td>0.75</td>
<td>0.46</td>
<td>0.57</td>
</tr>
</tbody>
</table>

Table 2. The Performances of Naïve Bayes Classifiers

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Recall</th>
<th>Precision</th>
<th>F-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>0.15</td>
<td>0.71</td>
<td>0.25</td>
</tr>
<tr>
<td>-9</td>
<td>0.18</td>
<td>0.68</td>
<td>0.28</td>
</tr>
<tr>
<td>-8</td>
<td>0.22</td>
<td>0.65</td>
<td>0.33</td>
</tr>
<tr>
<td>-7</td>
<td>0.27</td>
<td>0.62</td>
<td>0.38</td>
</tr>
<tr>
<td>-6</td>
<td>0.31</td>
<td>0.59</td>
<td>0.41</td>
</tr>
<tr>
<td>-5</td>
<td>0.36</td>
<td>0.56</td>
<td>0.44</td>
</tr>
<tr>
<td>-4</td>
<td>0.41</td>
<td>0.53</td>
<td>0.46</td>
</tr>
<tr>
<td>-3</td>
<td>0.46</td>
<td>0.50</td>
<td>0.48</td>
</tr>
<tr>
<td>-2</td>
<td>0.51</td>
<td>0.48</td>
<td>0.49</td>
</tr>
<tr>
<td>-1</td>
<td>0.55</td>
<td>0.45</td>
<td>0.50</td>
</tr>
<tr>
<td>0</td>
<td>0.63</td>
<td>0.40</td>
<td>0.49</td>
</tr>
<tr>
<td>1</td>
<td>0.66</td>
<td>0.37</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Table 3. The performances of raw confidence scores

Table 2 shows that for dataset I, NB classifiers predicted 75 percent of errors. And among the predicted errors, 42% were truly errors. For dataset II, 75 percent of errors were caught, and 46 percent of those predicted errors were truly errors. The F-Measure for the two datasets reached 0.53 and 0.57, respectively. Table 3 (extracted from [12]) summarized the classification result using raw confidence scores for dataset I. When the threshold was set to –1, the best F-Measure 0.50 was achieved. The measurement results were 55% for recall and 45% for precision. Therefore, the NB classifier achieved significantly higher recall than the raw confidence score approach.

The comparison of the results in Table 2 and Table 3 showed that NB classifiers produced significantly better results than the raw-confidence score approach, which supported our hypothesis. It also revealed that the latter was highly sensitive to the range of confidence scores.

4.4 Discussion

This study shows that a significant portion of errors can be detected in SR systems by using NB classifiers. This study innovatively applied a loosely connected approach to identify errors based on the generic information from SR output. The classifiers produced remarkable results of detecting about 75% errors in both speech corpora with over 41% precision. The results were significantly better in comparison with raw confidence score-based approaches from a prior study.

This paper questions a general belief about confidence score in the SR community that confidence score, as a kind of post hoc measurement, is of no use once the SR engine generates the output. The logic is whatever information contained in the confidence score has been used to decide the hypothesis. And then any play with confidence score afterwards is trivial. But this paper demonstrates that the confidence score, when used by a different approach and combined with other features, can still be quite useful.

Even though simple NB classifiers are not generally considered to be accurate models, they generated promising results on the test corpora in this study. This should not be surprising after a close examination of SR output. In terms of error detection, errors are clearly a minority in comparison with correctly recognized words. NB models are robust and are favorable to smaller population classes [37]. The highly dynamic nature of SR systems and the task of error detection may fit in with the strength of NB classifiers.

In this study, we assumed that misclassification of both false positives and false negatives were equally important. Since missing an error could be more serious than double-check a correct recognition, a higher weight may be assigned to false negatives in future.

It is important to note that detecting SR errors can be considered as a plug-in process to the recognizer. any classification problem, the selection of a decision threshold is a delicate question. We would like to have a plug-in model whose parameters depend as little as possible on the application. For example, in order to support users in error navigation (e.g. [12]), the threshold is more likely to be selected in favor of false positives.

The performance of classifiers may be improved by simply increasing the number of input features. Linguistic information such as parsing features generated by natural language processing could be very useful for identifying errors caused by ungrammatical structures.

In practice, it is relatively rare for a single ranking classifier to dominate all alternatives. Thus, other machine learning algorithms should be explored to find the most effective alternative. Furthermore, the classification performance can be improved by combining several ranking classifiers into a new, hybrid classifier that outperforms any of the individual classifiers [38].
Since error identification is imperfect, this information can only be used as decision support tools in many applications. A user must also examine words classified as errors by the system. The inability to automatically catch all recognition errors without false alarms suggests that error detection models should be combined with user verification and effective user support in order to achieve effective navigation. Similarly, the user may also randomly examine words classified as correct by the system.

There are many practical implications of the findings in this study. The models developed in this study can be incorporated into error correction systems to improve the specification of anchors during users’ navigation process. Moreover, they can also be integrated into speech recognition systems to improve accuracy of the SR output. Finally, the usability of SR systems, especially for people with disabilities, will be significantly improved.

5. Conclusions and Future Research

In this paper, we presented NB classifiers that take into account confidence scores, alternative hypotheses, contextual information, and domain features in order to identify errors in SR output. The experimental results demonstrated the superiority of NB classifiers as compared to a raw confidence score based approach for error detection.

This paper makes both methodological and application contributions. A unique feature of this paper is that only information available from the SR output was selected as input features to NB models. Moreover, contextual differential scores were innovatively selected to predict errors, which have shown positive impact on the classifiers’ performance.

It was shown that Naïve Bayes classifiers excelled in the recall rate. In the context of error detection, error recall is likely to be the greatest concern to most users of SR systems (as they would not want their documents contain undiscovered errors), which can be reflected in the asymmetric notion of cost used for classification. Thus, we suggest utility modeling with cost sensitive classification in the next step. Toward this end, we may assign higher costs to false negatives than false positives.

Error correction can be either automated or manual. The latter approach is less efficient, but more accurate. A good compromise could be to develop supporting system environments that can incorporate error detection classifiers, as introduced in this paper, to assist users in efficient navigation and error correction. We plan to present the SR results to users by highlighting identified errors and test how effective these error detection classifiers are in assisting users to navigate to errors.

By focusing on information that is available from normal SR output, we expect our findings to be generalizable to other systems and applications.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant# IIS-0328391. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

References


