Speak Correct: Phonetic Editor Approach

Hassanin Al-Barhamtoshy[1], Kamal Jambi[1], Wajdi Al-Jedaibi[1], Diaa Motaweh[2], Sherif Abdou[3], Mohsen Rashwan[4]

[1]Faculty of Computing & Information Technology; King Abdulaziz University, Saudi Arabia.
[2]Faculty of Education, King Abdulaziz University
[3] Faculty of Computers at Cairo University Egypt
[4] Faculty of Engineering at Cairo University Egypt
hassanin@kau.edu.sa

Abstract: In this paper, phonetic editor system for learning English speaking will be introduced. Methods and the architecture of systems used to edit new lessons into proposed dictionary will be discussed taken into consideration pronunciation effects. *Speak Correct* system will be presented, which uses state of the art automatic speech recognition (ASR) and examines pronunciation errors by speech engine. Two levels of teaching will be implemented; consonants and vowels, which are important for speech recognition. The two levels cover detailed accent defects that describe such articulation. The core engine of the Speak Correct was trained using prerecorded 100 hours of speech and used these data to create a pronunciation-training database. The proposed editor framework is optimized to suit an embedded phonetic pronunciation database and is useful for analyzing and detecting speech errors in Arabian region. The objectives of this paper is designing, implementing, and testing a prototype system that can add, and edit additional phonetic topics to cover pronunciation errors in teaching-based activities for adult students. In addition, the system will help teachers impart basic reading skills to assist students in comprehensive development.

[Hassanin Al-Barhamtoshy, Kamal Jambi, Wajdi Al-Jedaibi, Diaa Motaweh, Sherif Abdou, Mohsen Rashwan. **Speak Correct: Phonetic Editor Approach.** *Life Sci J* 2014;11(8):626-640] (ISSN:1097-8135). http://www.lifesciencesite.com. 93

Keywords. Phonetic Editor, Speech recognition, English vowels and consonants, Arabic dialects, acoustic error.

1. Introduction

Most speech engines are composed of word recognition, and phoneme recognition, and may use a variety of models, such as the Hidden Markov Model (HTK), to convert the given utterance into a sequence of phonemes. This sequence is processed using matching algorithms and the most important keywords are extracted. The acoustic input O is treated as a sequence of individual "*symbols*" or "*observations*", represented by symbols: $O = o_1, o_2, o_3, \ldots, o_t$. Similarly, a sentence/word will be treated as a string of words/phonemes: W = $w_1, w_2, w_3, \ldots, w_n$[1-4]. The general terminology used throughout this document is explained in the following sections.

1.1. Literature Review and Related Works

"Spoken Language Understanding (SLU)" and "Natural Language Understanding (NLU)" are systems that interpret signs conveyed by a speech signal [2]. SLU and NLU can be used to expand the conceptual representation of sentences in a natural language [2]. In such system, SLY and NLU interpret signs and code them into signals with additional information. Furthermore, such system includes an Automatic Speech Recognition (ASR) module that is sensitive to noise, according to the nature of the spoken language and the errors introduced by ASR.

Dialog classification and automatic segmentation are central to SLU. Another paper proposed a framework for contextual speech, extract prosodic features to segment, and classify meetings[5]. They reported that: "contextual features are better for recognizing, while prosodic features are better for finding base mechanisms and backchannels" [5].

In a study by [6], voice inputs were compared to those in a database (2009). The authors presented a phonetic similarity technique [6] that had been applied in the music domain. They found that search mistakes were minimized both in text and spoken queries [6].

In a paper by Heracleous, consonant and vowel recognition exercises were presented in French using Hidden Markov Models (HMM). The hand-shapes and lip-patterns of speech (as a visual communication mode) strongly contribute to meaning in oral languages, especially for deaf and hearing-impaired people. Thus, the objective of their research was to address difficulties in lip reading, in an attempt to enhance language comprehension in deaf children and adults [7].

Two approaches have been used to increase speech intelligibility for speaking impaired people [8]. The first approach is related to the context of conversation for hearing-impaired listeners; the second approach aims to raise the intelligibility of speaking-impaired persons. To date, an intelligibility increase has not been achieved, and listeners preferred to listen to transformed speech produced by an alternative system.

Linguistic knowledge is generally used in ASR to improve error prediction [9]. In Tsubota, 79 pronunciation mistake patterns were modeled for English spoken by Japanese students [10]. This was done using a simple approach: the experimenters followed the pitch of two active speakers, and applied HMM to track pitch over time. They then used a statistical model to demonstrate their experimental results. The paper showed remarkable performance of the proposed process in comparison to a multi-pitch tracking algorithm.

A Computer Assisted Language Learning (CALL) model has been developed to assist students learning Japanese [11]. The proposed model perceives lexical and grammatical errors in pronunciation, as well as input sentence errors. Additionally, a method has been proposed to generate acoustic sub-word units from a spoken term detection system that can be substituted for conventional phone models [12]. The system generates a set of speaker models in an unsupervised method that was exclusively designed for language training. Another paper [13] describes an error classification decision tree that can be used to find critical and redundant errors in automatic speech recognition.

One paper [14] focused on the topic of non-native accents in continuous speech recognition. The authors proposed a system for analyzing the transformation rules of non-native Mandarin spoken by native speakers**.** They used the Mandarin speech corpus to train HMM models to test speech recognition performance. Their results were positive in that they obtained information about adapting a native speaker ASR system to a model with nonnative accented data.

A paper titled “Vowel Effects towards Dental Arabic Consonants based on Spectrogram” [15] discussed the effect of Arabic vowels on Arabic consonants using diacritics interpreted by Malaysian children. Vowels were added to the essential consonants with three simple diacritics. The paper reported that the location of articulation is important for dental consonants and formant frequencies.

In a study by Kensaku [16], the authors cancelled acoustic echo by substituting the difference between coefficients of an adaptive filter for the estimation error. Additionally, Abdou and et al discussed the impact of "Speak Correct" system which is a Computer Aided Pronunciation Training (CAPT) system for native Arabic students in teaching English. Evaluation results for the system are promising and show significant improvements in the users' pronunciation proficiency.

1.2. Accent Defects and Pronunciation Error Categorization

While some errors in speech may be noticeable without inhibiting understanding by a native listener, other types of errors may cause serious problems for comprehension of nonnative speakers. Regardless of overall comprehension goals, any learner will benefit from realizing the impact of various errors.

Based on previous studies, we investigated pronunciation difficulties in individuals from Egypt and Saudi Arabia who were learning English as a second language. Our main motivation was to create guidelines for teaching English as a foreign language. For our analysis, we sought to obtain recordings that were representative for the learner group and that covered all aspects of pronunciation. The phonological descriptions of the recorded data are as follows.

General Phonology.

The phonological systems of the Arabic and English languages are different, especially in terms of the range of sounds used in vowels and consonants. Twenty-two vowels exist in English with 24 diphthongs and consonants, while Arabic has only eight vowels and diphthongs and 32 consonants [20]. Arabic vowels include three short, three long, and two diphthongs. Therefore, Arabic speakers gloss over and confuse short vowel sounds in English, emphasize consonants, and avoid elisions and shortened forms [14, 15, and 20]. There exist a wide variety of dialects within each Arabic country. Thus, it is necessary to consider differences in pronunciation and language structure [20].

When completing a speech task, there are many sources of pronunciation errors and acoustic variation in Saudi and Egyptian accents. Using an analysis of acoustic [20] sorted errors, the “S*peak Correct”* team constructed an intelligibility scale as a guideline for prioritizing aspects of pronunciation during language instruction. The most serious errors and the initial work on creating pronunciation error detectors for the proposed framework are described in the following subsection.

For spelling error detection, we looked for defects in pronounced text, which can inhibit correct spelling. In the following subsection, we break the field down into four increasingly broad problems:

1. **Substituting:** /v/ for /f/, such as saying: vat/fat, very/ferry, belief/ believe, vast/fast, and van/fan.
2. **Rolling:** the /r/ for example: Library, Ruler, Lorry, Liberian, and Reroofing.
3. **Replacing:** /θ/ with /s/, as in sin for thin, thong/song, thank/sank, theme/seem, thin/sin, and thought/sought.

4. **Dental Fricatives**: /θ, ð /, Replacing /ð/ with /z/ or /d/, as in dat or zat for that, and /ð/ with /θ/. Therefore, Ss may replace the /ð/ sound as in “brother”, “they”, and “these”, with the /θ/ sound. For example: another, blithering, bother, brother, and father.

Consonant Clusters.

The number of consonant clusters that occur in English is greater than that in Arabic. For instance, initial two segment clusters rarely occur in Arabic [20]: pr, pl, gr, gl, thr, thw, and sp. Initial three segment clusters do not occur in Arabic: spr, skr, str, and spl. According to these clusters, there is tendency to insert short vowels to support pronunciation (among Arabic speakers): ispring or sipring for spring, and perice or pirice for price. This also occurs in the range of final clusters [20]: monthiz for months and neckist for next. Most of these pronunciations can be categorized as **insertion** and **deletion**.

2. Common Pronunciation Errors

Figure 1 illustrates pronunciation errors that need to be addressed in successful training and assessment models [24]. As shown in the figure, such errors can be classified into phonemic and prosodic types.

(1) Phonemic errors -in this paper- can be categorized based on whether they are substituted, deleted, or inserted. Also, small- scale errors occur “where the correct phoneme is more or less being spoken” [24].

(2) Prosodic errors can be categorized based on whether they involve stress, rhythm, or intonation.

These two types of errors make pronunciation a multi-dimensional problem. Consequently, a large numbers of metrics are used to measure these dimensions [24, 25].

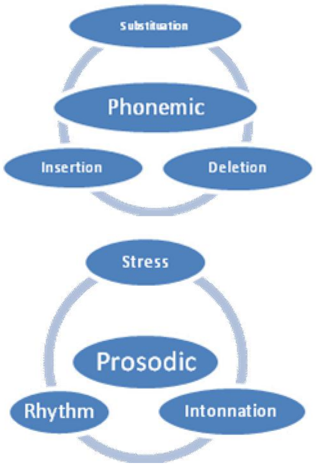


Figure (1): Classification of Pronunciation Errors

During the development of Speak Correct, we found a significant body of literature describing typical patterns of error: Korean Learner Segmental Errors (KLEs) [26]. A pilot corpus was collected and phonetically annotated using prompted English speech data from several different types of content. The corpus included short paragraphs of text, sentence prompts, and words with particularly difficult consonant clusters (e.g., refrigerator). In total, the pilot corpus included 25,000 speech samples from 111 learners who resided in Korea. The corpus provides a direct comparison between realized phonetic sequences and expected canonical sequences from native speakers. This summary of common errors made by Korean speakers learning English indicated that these speakers do not make a distinction between fricatives, /f/ and /v/, and substitute /p/ and /b/ instead. Other common errors include the substitution of aspirated /t/ for /θ/ and un-aspirated /t/ for /ð/ [24]. Another study illustrates the most frequently observed segmentation errors [26]. The current paper focuses on phonemics substitution, deletion, and insertion of errors.

3. Speak Correct Phonetic Editor

The proposed system of the Speak Correct has four main stages; trainer stage is used to train speeches features, decoder stage is used for speech decoding with pronunciation hypothesis, evaluation stage to evaluate &generate speaker's feedback, and the phonetic editor is organized to enrich the proposed language model with mapping and generation rules adaption. The system uses a decoder that recognizes user input speech, measures confidence and pronunciation error. In addition, the system includes analysis of feedback messages and detection of errors, guidance is given to correct the errors, and an evaluation takes place. Phonetic editor includes mapping and generation rules with statistical techniques (e.g., neural networks or Gaussian models) to recognize individual speech sounds.

The data flow diagram (DFD) of the proposed *Speak Correct*is illustrated in figure (2). Each word is displayed to the users with graph representation and lattice form, with a picture representing certain phonetic levels and related lessons (Saudi or Egyptian accent defects). Thus, the *SpeakCorrect* allows users the freedom to select their own levels and examples.

At the phonetic stage; the utterances are entered via expert annotators, mapping and generation rules with automatic speech recognition are employed, which is supported by trained instances, in the form of a grammar network (Lattice graph) for the target word. Errors are detected and analyzed, and the evaluation takes place using the decoder and evaluation stages.

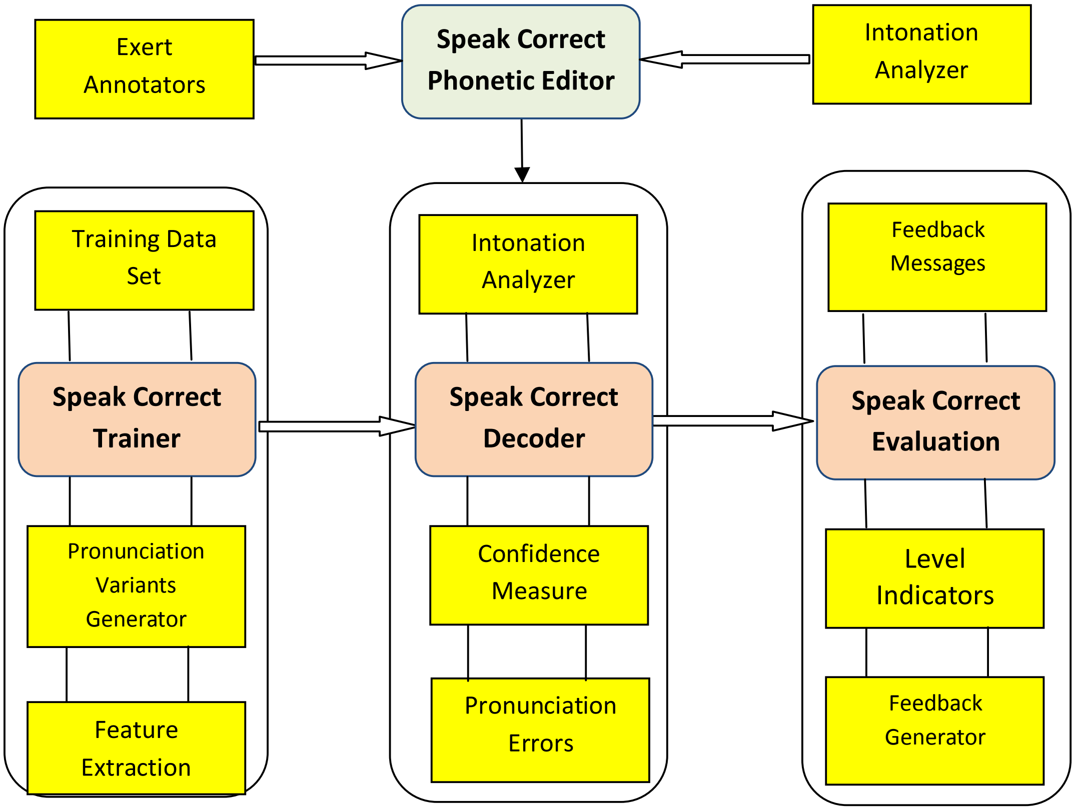


Figure (2): Speak Correct with Phonetic Analyzer Overview

The *decoding* stage addresses the problem of finding the correct "*underlying*" sequence of symbols/ patterns. Therefore, the Veterbi algorithm is an efficient way of solving the decoding problem by considering all possible strings and using addition rules (such as Bays rule [20]) to compute the probabilities of generating the observer sequence.

Further processing is often performed to adapt the reference speech models to the speaker speech properties. In such cases, the Maximum Likelihood Linear Regression (MLLR) speaker adaption module is used to refine the adapted module.

3.1. Acoustic Probabilities Counting

As previously mentioned speech input can be passed through signal processing transformations and converted into a series of feature vectors, where each vector represents a time-slice of the speech input signal. A popular way to compute probabilities for feature vectors is to first cluster the feature vectors into discrete counted symbols. The probability of a given cluster can then be calculated (based on the number of times it occurs in a training set).

This methodology is called ***vector quantization***, and is derived from either computing observation probabilities or probability density function (pdf). There are two common approaches: **Gaussian pdfs** can be used to map the observation vector $\boldsymbol{O_t}$to a probability, and neural networks or multi-layer perceptions can be used to assign probabilities to real-valued speech feature vectors. The neural network is a set of small computational units connected by weighted links. The network is given vector values and computes a vector of output values.

A standard model based on a probabilistic neural network is proposed in.[21]; this model is suitable for testing and pattern classification. The structure of the probabilistic neural network model includes the number of input speech variables M, the number of identification patterns needed N, and the training samples for each pattern are represented by S_1, S_2… S_N. Four layers exist: the input layer, model layer, summation layer, and output layer, and the weights between the summation layer and output layer are computed by:

$$W(M) = Si / \sum_{i=1}^{N} Si$$

Consequently, when speaker adaption takes place, speaker features related to acoustics, gender, accents, and age will be modeled in speech processing.

Therefore, a speaker's unique accent should not be affected.

3.2. Speak Correct Editor and Phonetic-based Approach

The transformation rules of the Speak Correct system comprise a phonetic-based process that presents text words as pronunciation words. Therefore, the system uses the intermediate form between the source words and target words, based on a rule of phonetic-translation that captures the pronunciation of the target words. Three phonetic-based processing rules are used: identification, mapping, and generation rules. The identification rule processes phonemes in the source word(s), the mapping rule represents the association of those phonemes to characters represented in the target word(s) (orthographic representation), and the generation rule generates the target word as a pronounced word (letter-to-sound rule).

The transformation rule concepts are based on the following model:

$$P(W_s , W_t) = P(W_t) \sum P (W_s \mid I_s) P (I_s \mid W_t)$$

where $P (W_s \mid I_s)$ is the probability of pronouncing the source word; $P (I_s \mid W_t)$ is the probability of generating the written W_t from the pronunciation in I_s ; and $P (W_t)$ represents probability of sequence W_t occurring in the target language.

The HMM or ATN can be thought of as a transformation rule with the source input (W_s) and mapping of the target output (W_t) using the weight for each transition between states. Therefore, the transformation rule specifies which output sequences have highest probability. With respect to the target language, $P(W_t)$ is a unigram word model and can be implemented using any corpus. $P (W_s| I_s)$ can be estimated based on frequency information.

3.3. Speak Correct Principles Modules

Our goal was to build a model, figure out how it modified a "true" word, and then recover the word. For the complete speak correct tasks, the basic recognition processes are as follows.

1) Main Module

Step 1.1: Gathering speech input samples.

Step 1.2: Dividing the samples into two parts, such that one part is for training and the second part is for testing.

2) Training Module

Step 2: Do the following steps to train the *Speak Correct* model:

2.1 Speech Adaption.

2.2 Confidence measurement.

2.3 Tuning the accent of the native Arabic speaker.

a. Tuning Saudi accent

b. Tuning Egyptian accent.

2.4 Intonation and pronunciation training.

3) Phonetic Editor Module

Step 3: Do the following steps to accept and enrich the *Speak Correct* model

Step 3.1: Select the level of the lesson with the predetermined acoustic and language model.

Step 3.2: Add the new lesson with the selected level.

Step 3.3: Add the words within the added lesson inside the level.

4) Testing and Evaluation Module

Step 4: Do the following steps:

Step 4.1: Establish the system with the associated acoustic and language models.

Step 4.2: Use the feature vectors to input test samples into the trained network.

Step 4.3: Judge the equivalent speech signal and the speaker characteristics according to the output values.

Step 4.4: Evaluate the speakers according to the feedback messages.

3.4. Weighted Finite State and Weighted ATN/Lattice

Computational linguistics and automata theory have been used to predict letter sequences, describe natural language, employ context-free grammar (CFG), introduce the theory of tree transducers, and parse automatic natural language text [29-37]. In the 1970s, speech-processing researchers captured NLP grammar with weighted Finite State Acceptors (FSAs), utilizing transition weights that could be read by computers with access to dictionaries, corpus, and corpora [36, 37, 38-44].

In the 1990s, finite state machines and large training corpora became the central model in speech processing, and software toolkits for Weighted Finite State Machines (WFSM) were developed [29]. Finally, the 21st century has seen the development of common tree automata toolkits [36, 37] to support investigations.

The single WFST or Augmented Transition Network (ATN) that represents P(S|E) is complex, although model transformation can produce a chain of transducers in the following manner:

$$WFSM_a(English_{text}) \leftrightarrow WFSM_b (English_{sound})$$

Therefore, a simple model can be used to calculate 1-gram, 2-gram, and n-gram language models of characters [29]. For instance, if a corpus includes 1,000,000 characters and the letter e occurs

127,000 times, the probability P(e) can be estimated as 0.127.

A 2-gram model can be calculated by remembering the previous letter context, otherwise known as its WFSA state. For example, in the transition between state s and state e, the letter e can be calculated by the probability P(e|s). The n-gram model generates more word-like items than the (n-1)-gram model. The weighted or lattice model is a simple automaton in which each arc is associated with a transition, this transition can be represented by a probability value indicating what path will be taken. The probability of all arcs leaving a node must sum to 1. Figure 3 shows a weighted ATN for the English word *"about",* which is trained on an actual pronunciation example. This model is an instance of a Hidden Markov Model (HMM). Such figures graphically illustrate the behavior transition in the weighted ATN. The rule of the transition is as follows:

- Starts in some initial state (start: s_1) with probability $p(s_i)$,
- On each move, goes from state s_i to state s_j according to transition probability P (s_i, s_j).
- At each state s_i, it emits a symbol w_k according to the emit probability P'(s_i, w_k).

The *Speak Correct* system is a hybrid approach, since it uses elements of the HMM or weighted state-graph representation of the pronunciation of a word, as well as the observation-probability computation based on multilayer perception. The network has one output unit for each phone, and by summing the values of all output units to 1, the *Speak Correct* can be used to compute the probability of a state j given an observation vector O_t, $P(q_j \mid o_t)$, or $P(o_t \mid q_j)$. Therefore, when receiving a sequence of spoken words that produce a given type of auditory speech, a standard model - like that described in [20] - is used. The model generates P(E|S) for a received speech signal S, as follows:

1. For each phonetic in S, a sequence of phonemes is observed with varying probabilities, and therefore can be interpreted as a word.
2. For each phonetic, a word-to phone transition is constructed.
3. Each phone can be expressed as a variety of acoustic signals.

Once defined, the chain of audio signal and the final language model are weighted with the method of likelihood, and the observation probabilities from the training data.

3.5. Training the Speak Correct

Here we present a brief sketch of the embedded training procedure that is used in most ASR systems. Some of details about the algorithm have been previously introduced [8, 11, 16, 21, and 22]. Four probabilistic models are needed to train the *Speak Correct* system:

- Language model probabilities: $P(w_i|w_{i-1}\ w_{i-2})$
- Likelihood observation: $b_j(o_t)$
- Transition probabilities: a_{ij}
- Pronunciation Lexicon: Lattice or Weighted ATN of the HMM state graph structure.

To train the previous probabilities component, the *Speak Correct* has the following corpuses:

- Training corpus of speech wave files: these were collected from news web sites on the internet, individual people etc. These speech wave files were collected together with the speech transcriptions.
- Large corpus of text: including the transcriptions from the speech corpus together with many other similar texts.
- Smaller training corpus of speech: which is phonetically labeled, i.e. frames are hand-annotated with phonemes.

The HMM lexicon structure is built using an off-the-shelf pronunciation dictionary. Therefore, the training begins by running the model and observing which transitions and observations were used. Any state can generate one observation symbol; the observation probabilities are all 1.0. The probability p_{ij} of a particular transition from state i to state j can be determined by calculating the number of transitions that occurred; c (i→ j), then normalizing such values using the following:

$$a_{ij} = c\ (i \rightarrow j) / \sum_{q \in Q}^{\infty} (C\ (i \rightarrow j)$$

Two methods can be used for the lattice or weighted ATN and HMM. The first is to iteratively estimate the counts and observation probabilities, and then use the estimated probabilities to derive better and better probabilities. The second involves obtaining estimated probabilities by computing the forward probability among all different paths. For instance, one can define the forward probability in state i after seeing the first ***t*** observations, given the automaton ***A***.

$$a_t\ (i) = \ P\ (o_1 , o_2 , o_3\ , \ldots , o_t\ , q_t\ = i \mid A)$$

Formally, the following iteration can be defined based on:

1. Initialization:

   $\alpha_n(1) = a_{1j} * \ b_j\ (o_1)$ ………. 1 < j< N

2. Iteration:

   $\alpha_j(t) = [\ \sum_{i=2}^{N-1}\ \ a_i\ (t-1) * a_{ij}\ ]\ b_j\ (o_t)$
   ……… 1 < j < N, 1 < t < T

3. Termination:

$$p(o|A) = a_N(T) = \sum_{i=2}^{N-1} a_i(\mathrm{T}) * a_{iN}$$

The speak-correct algorithm can be run to compute the candidate phonemes that were most probable given the observation sequence [ax b], such that the product P(o | w) P(w) is computed for each candidate word. Thus, the likelihood of observation sequence *o* given the word *w* times the prior probability of the word is computed for each word and the word with the highest value is selected.

Such algorithm is an edit distance algorithm; an intermediate table is used to store the probability values of the observation sequence. The data are represented in the table in rows, which are labeled by state-graph. The table is filled as a matrix, by computing the value of each cell from the three cells around it. Furthermore, the algorithm computes the sum of probabilities of all possible paths that could generate the observation sequence. Formally, each cell expresses the following probability:

$$\textit{speak}\,[t, j] = P(o_1, o_2 \ldots o_t, q_t = j \mid A)\, P(w)$$

The following pseudo code describes the speak algorithm applied to any word.

```
speakAlgorithm( observation, state-graph )
begin
    ns = numOfStates(state-graph);
    no= length(observation);
    /* create probability matrix */
    speak [ ns+2 , no + 2 ];
    speak[0,0] = 1.0;
    for each time step t from 0 to no do
        for each states from 0 to ns do
            for each transition s' from s specified by state-graph
                speak[ s' , t +1 ] = forward [ s , t ] * a[s , s'] * b [s', $o_t$];
    return sum of the probabilities in the final column of forward;
end.
```

where:

a [s , s'] represents the transition probability from the current state s to next state s'.

b [s', o_t] is the observation likelihood of s' given o_t.

b [s', o_t] is equal to 1 if the observation symbol matches the state, and is equal to 0 otherwise.

3.6. Data Set of Speak Correct

The first dataset we used contained information from two different domains. The first domain involved speech recordings collected from the Al Jazeera online news website. This dataset included around 140 recorded hours. We used 100 hours to build the *Speak Correct* language model, and around 40 hours to test the *Speak Correct* system. The second domain was divided into two regions: Saudi and Egyptian accents. Table 1 presents the structure of our dataset after recording.

Table 1: Structure of the dataset

Dataset 1	Al-Jazeera news	
Type	Training	Testing
No of hours	100	40

Dataset 2	Saudi		Egypt	
Type	Male	Female	Male	Female
No of students	39	-	40	30

In dataset 1, both the training and testing dataset were taken from native speakers. For dataset 2, we noticed that while annotating, we encountered some difficulties with interpretation, which may have been due to the following factors: regularities to record at the female section under the supervision of acoustic and linguistic male member.

3.7. Similarity between two English Words

Given two word phonemes, $W_1(p^1\ p^2\ p^3\ \ldots\ p^n)$ and W_2 $(p^1\ p^2\ p^3\ \ldots\ p^n)$, three factors are used to describe and evaluate similarity:

1. The similarity of pronunciation in each phoneme pair ($P^i(W_1)$, $P^j(W_2)$) between W_1 and W_2.
2. The similarity between the length of W_1 and the length of W_2, where 0 <= i<=n, 0<=j<=m.
3. The similarity between each character pair between W_1 and W_2.

The three factors have different roles in calculating the similarity between two words, as follows:

$$S_{w1}(W_2) = \sum_{i=1}^{3} w_i S_{iw1}\,(W_2)$$

In our case, w_1 was selected to be 0.5, w_2 was 1, and w_3 was selected to be 1.

3.8. Speak Correct Error Correction

To produce a correct utterance sequence, a kernel feature model matrix was used to calculate the similarity between two words (Syllable words). Given two words W_i and W_j, the confusion score between W_i and W_j can be calculated as the average confusion of all speech segments S_i annotated as W_i in the trained HMM model for W_j,
A_{wj}.

Consequently, the confusion similarity score for W_i and W_j can be estimated using the following equation:

$$Sim(W_i \text{ and } W_j) = (P(O_i|A_i) + P(O_j|A_j))/2$$

The Guassian kernel function can be applied to calculate the confusion score between W_i and W_j. $Conf_{Score}(W_i , W_j) = \exp ((Sim (W_i , W_j))^2 / 2 \sigma^2)$ whereσ represents the variance calculated over the distribution Sim (W_i , W_j).

The best correction result can be calculated according to the following equation:

$$C= \text{argmax} (P (W) P (E | W) P (W|S))$$

where P(W) represents the word language model for the corrected word sequence W. Figure 3 illustrates the proposed *SpeakCorrect* correction system.

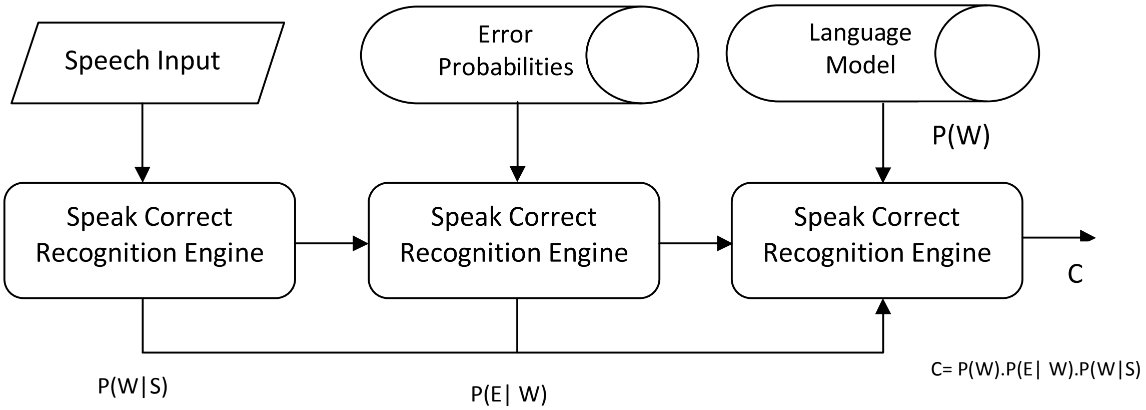

Speech Input
Error Probabilities
Language Model
P(W)
Speak Correct Recognition Engine
Speak Correct Recognition Engine
Speak Correct Recognition Engine
C
C= P(W).P(E| W).P(W|S)
P(W|S)
P(E| W)

Figure (3): Speak Correct Error Correction

4. Speak Correct with Phonetic-based Implementation

In the following section, we describe the components of our model related to pronunciation analysis and pronunciation adaption.

The *SpeakCorrect* corpus is based on annotated speech; with the intention of providing acoustic information to support the development and evaluation of automatic speech recognition systems.

Like the Brown Corpus, ***Speak Correct*** includes a balanced selection of dialects, speakers, and materials. It contains data from two main accent regions with two dialect localities for each. 150 male and female speakers (ranging in age from 18-21 years) with undergraduate educations each read 390 carefully chosen words. We chose phonetically rich words that covered all the pronunciation defects (substituting, deletion, and insertion) observed in Arabic speakers (Saudi and Egypt regions). Our design required multiple speakers to say the same words to permit comparison across speakers, and a large range of words was necessary to obtain maximal coverage of defects. We obtained 15,000-recorded utterances, which we stored in the corpus. Each file name has internal structure, as shown in figure 5.Each item has a phonetic transcript, which can be accessed via the corresponding word tokens.

Speak Correct includes several corpus design features, as shown in figure 4. First, the corpus contains annotations at the phonetic and orthographic levels, with different labeling schemes at each level. Additionally, there are multiple dimensions of variation, to cover accents, dialect regions, and localities, thus facilitating the use of the corpus for sociolinguistic research.

The user interface in ***Speak Correct*** is divided into three tiers. The top part contains the presentation tier; the middle part includes the logical or business tier; which starts with registration, where login takes place, the user adjusts microphone settings and preferences, the language and speech lessons, and finally the evaluation. The third tier is internal, and hosts all the properties, databases, files, etc, for the program.

The user interface was designed using Silverlight technology. This user interface includes different visual properties for basic functions, such as moving between demos, playing a sample (predefined example), testing the user voice, and recording user voice. Figure 5 illustrates the device setting and microphone adjustment interface.

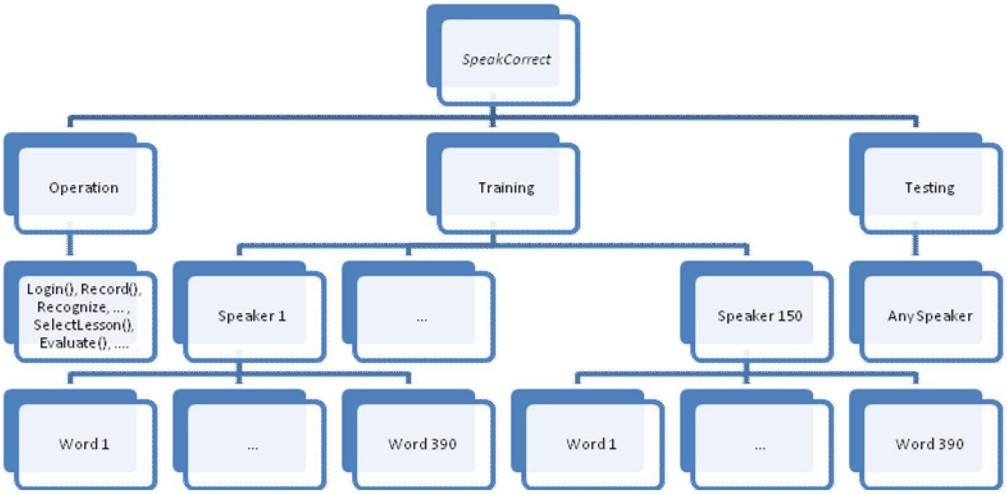
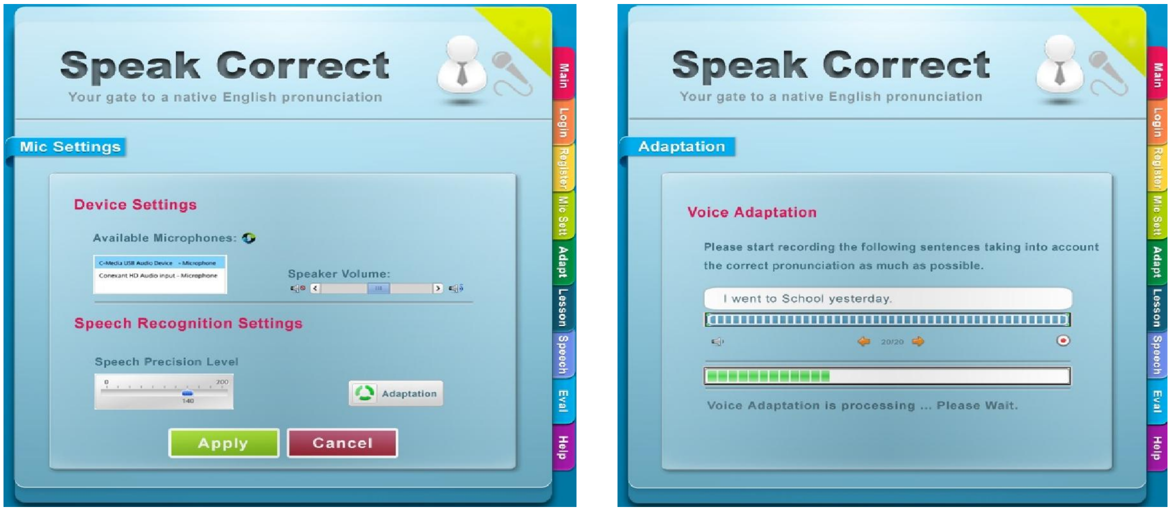


Figure (4): Structure of the Implemented Speak Correct Corpus



Device Settings

Available Microphones:

C-Media USB Audio Device - Microphone
Conexant HD Audio input - Microphone

Speaker Volume:

Speech Recognition Settings

Speech Precision Level

0 200

140

Voice Adaptation

Please start recording the following sentences taking into account the correct pronunciation as much as possible.

I went to School yesterday.

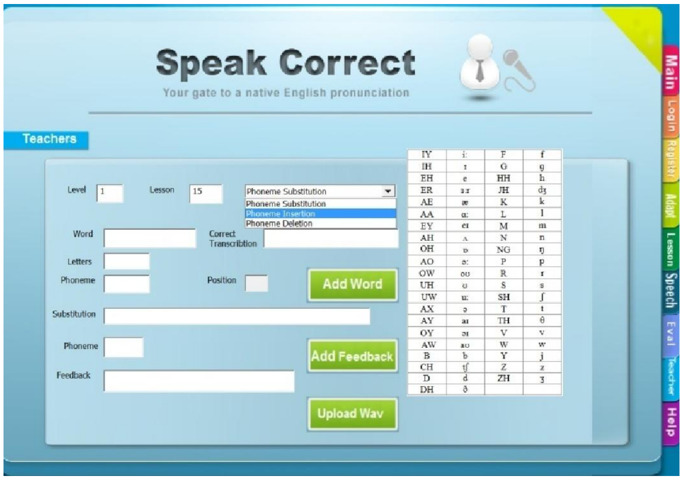
20/20

Voice Adaptation is processing ... Please Wait.

Figure (5): The Device Setting and Microphone Adjustment Interface of the Speak Correct System

The implementation code contains a collaboration module between C# code (.Net Client/Server), HMM, and the HTK component code. The second component was used to compare the input voice against the predefined trained voices, and therefore provide feedback.

The following testing model uses components to test, evaluate and guide students through pronunciation editing, analysis, adaption, and evaluate pronunciation errors with specific focus on acoustic accents. Figure 6 illustrates phonetic editor components. The user selects the level, lesson and the title of the type of errors (Substitution, deletion or insertion). Other features (word, sequence of phonemes, upload wave file) can be added using such editor.



Level 1 Lesson 15

Phoneme Substitution
Phoneme Substitution
Phoneme Insertion
Phoneme Deletion

Word

Correct Transcribtion

Letters

Phoneme

Position

Substitution

Phoneme

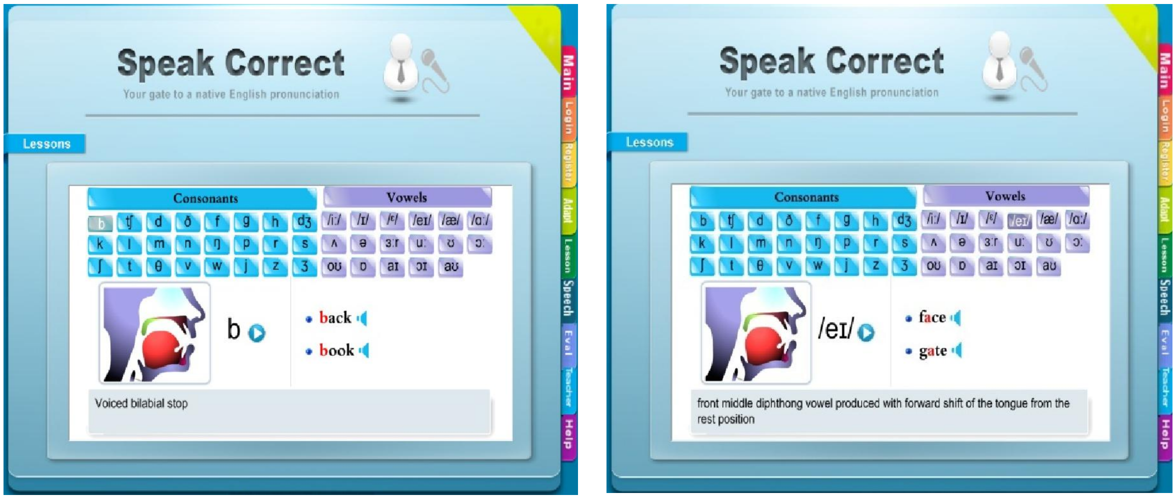
Feedback

IY	iː	F	f
IH	ɪ	G	g
EH	e	HH	h
ER	ɜːr	JH	dʒ
AE	æ	K	k
AA	ɑː	L	l
EY	eɪ	M	m
AH	ʌ	N	n
OH	ɒ	NG	ŋ
AO	ɔː	P	p
OW	oʊ	R	r
UH	ʊ	S	s
UW	uː	SH	ʃ
AX	ə	T	t
AY	aɪ	TH	θ
OY	ɔɪ	V	v
AW	aʊ	W	w
B	b	Y	j
CH	tʃ	Z	z
D	d	ZH	ʒ
DH	ð		

Figure (6): Proposed Phonetic Editor of the Speak Correct

5. Speak Correct Interactive Phonetic Editor Experimental Test

Adding new level, lessons and therefore related sequence of phonemes with correct pronunciation are serious problem for speech recognition systems, especially for non-native English speakers with accents. To address such difficulties, our phonetic editor system prompts users to say specific words and utterances, and then adds the phoneme sequences of the spoken words to enrich the built-in corpus and determine the user's accuracy. Therefore, the *Speak Correct* system must be trained to recognize every word based on the pronounced phonemes. First, a lattice graph of possible phoneme sequence is generated, and then the speech sample of the entry word is mapped at the recognition phase module. This is done via a search/match procedure performed with the lattice graph to identify the best matched phoneme sequence. The speak Correct system includes lessons about the predesigned levels that covers English consonants and English vowels [45], see figure 7.



Voiced bilabial stop

front middle diphthong vowel produced with forward shift of the tongue from the rest position

Figure (7): Speak Correct Predefined Consonants and Vowels Lessons

If a user was being tested for pronunciation of the English word "picture", which is pronounced /pik-cher/ or /pĭˈk chər/, and they pronounced the word according to their training (original frame sequence of the lattice graph), then the output would be correct.

According to some literature [38, 39, and 40], speech recognition systems are unreliable when using a phoneme recognition model, such that speech recognition has ~ 80% accuracy [39, 41]. We developed an interactive system that incorporates word utterances and sequences of phonemes. Using this technique, users could correct misrecognized phonemes based on wave graph responses. The origins of the utterance evaluations for the *Speak Correct* system are described below.

1. **Vowels-based Error Correction.** This proposed interaction module of the *Speak Correct* system enables users to display misrecognized phonemes according to vowels pronunciation errors. The system could first locate vowel errors, and then ask the user to attempt a correction after displaying the correct vowel sequences. During this interaction, the system would scan the pre-defined grammar dataset to determine the accuracy of the user's vowels utterances, and then signal the sound wave of such utterances.
2. **Consonants-based Error Correction.** The *Speak Correct* system enables users to display misrecognized phonemes according to consonants pronunciation errors. The system could first locate consonant errors, and then ask the user to attempt a correction after displaying the correct consonant sequences.
3. **History-based Evaluation.** The proposed *Speak Correct* system uses historical information regarding errors in phoneme sequences that were previously detected (corrected or not) to evaluate user performance.

5.1. Accent Utterances and Phoneme Errors in Speak Correct

A confidence measure of the Generalized Posterior Probability (GPP) [42] is used to calculate the reliability and subsequent matching of the users' utterances. Therefore, a phoneme distance measure is calculated from the phoneme confusion/matching matrix. This matrix was based on the stored Saudi and Egyptian accents database, which consist of 300 speech samples from 70 speakers for each region (35 males and 35 females), for a total of 21000 samples.

Nakamura et al., used a phoneme recognizer to build the confusion matrix [43].

The confusion matrix C (α , β) represents β number of phonemes recognized by α phonemes. The phoneme distance can be computed by [40]:

$$D(\alpha , \beta) = - \log \frac{C(\alpha ,\beta)}{\sum C(\alpha i ,\beta)}$$

Also, the GPP can be used to verify recognized sub-words, words, and sentences [42]. It is calculated by generalizing the likelihoods of the different entities (sub-word, word, or sentence). The relationship between recognition accuracy and the GPP for speech recognition has been previously investigated [40].

5.2. User History Evaluation in Speak Correct

The *Speak Correct* dataset includes utterances from 40 students. The students are female and male native Arabic speakers enrolled at the IT College. Each student was asked to attempt 10 examples for each lesson of the *Speak Correct* system. Examples were randomly selected to be included in the Speak Correct dataset (12756 utterances). The dataset can be categorized into two parts: calibration and evaluation.

Various details regarding user evaluation history can be displayed as sequences of examples {e_1, e_2... e_n} that were registered during previous during interactions with *Speak Correct*. While interacting with *Speak Correct*, the system ensures that the correction results for each phoneme sequence p_i are separated from the non-corrected phoneme sequences. The algorithm for user evaluation is shown below. Word-based corrections, uncorrected behavior, “Change the level of testing/ change related example”, and “stop/end” evaluation are four types of testing that can be employed in this algorithm. The initial decision “It is correct” is used to continue and to encounter new words. The second decision “It is not correct” is used to correct the utterance or pronounced phonemes. “Change the level of testing/ change related example” is used to change or select another level or example. The last decision, “stop/end”, can be used to terminate the *Speak Correct* evaluation algorithm.

Speak Correct Evaluation Algorithm

Do the following steps.

```
Begin
1.  Initial Step: level = 1; example = 1; correct=0;
    nonCorrect=0; m = maxNoOfExamples;
2.  Use the recognized phoneme sequences from the
    Speak Correct training module (Lattice Graph
    with learnt rules)
3.  Request user utterance entry (Level and examples;
    Speak the displayed word)
4.  According to the user responses, do the following:
    If the user utterance is recognized with the
    decision “is correct” go to Step 5
    If the user utterance is recognized with the
    decision “is not correct” go to Step 8; the entry
    selection has a phoneme error(s)
    If the user selects “Stop/End” go to Step 10
5.  correct = correct + 1
6.  If correct > m Then go to Step 10; Otherwise go
    to Step 4
7.  Use the recognized phoneme sequences $e_i$ of the
    word at index i; store it in the correct set of
    examples; go to Step 4
8.  nonCorrect = nonCorrect +1; store the tested
    example in the non-correct set
9.  According to the user’s response, do the
    following:
    If the user selects “Stop/End” decision, go to Step
    10
    If the user selects another Level/Example, go to
    Step 4
    Otherwise display the correct list and the non-
    correct list;
    Calculate the user Score
10. Stop/End
End.
```

The following section discusses experimental evaluation of the Speak Correct word pronunciation task. First, we evaluated the performance results of the *SpeakCorrect* system by comparing the obtained results, before and after the training module.

5.3. Speak Correct Evaluation Scenario

The *SpeakCorrect* training process is organized in 6 levels. Each level includes between 10 and 35 examples. The levels cover vowels, and consonants; with approximately 390 selective words. The *Speak Correct* points are spread across a set of teaching lessons, with an average of 160 lessons[1]. Each lesson consists of up to 15 words as examples. The words represent a collection of related words that often produce accent defects in non-native speakers. Before working through the lessons, users complete an overview of the levels and are exposed to related examples from the word data set.

5.4. Speak Correct Feedback

Error classification is achieved by comparing the features of the observed phoneme streams of a given word to those from the pre-trained word before using the *Speak Correct* system. Therefore, the system provides feedback about the mistakes made by users (students). The feedback includes error categorization

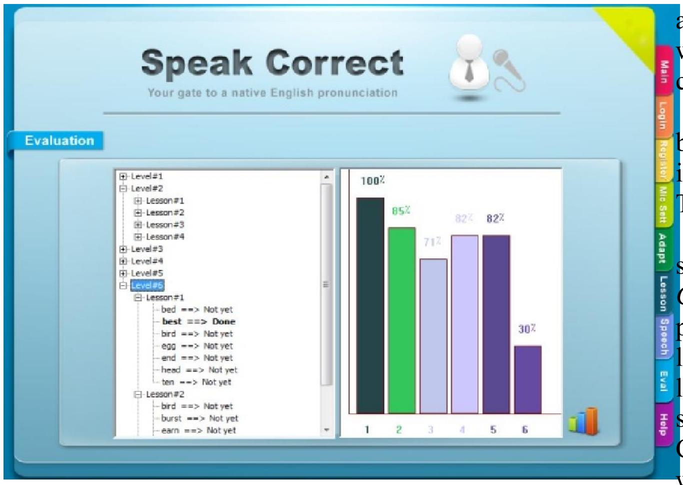
[1] The vowel level includes 15 lessons, and the constant level contains 5 lessons.

based on the number of errors at each level, within the examples.

Feedback in the *Speak Correct* system is provided through vocalized words and visual features. Students can select the lessons from a levels list, and can read the pronunciation of any lesson words. Also, students can listen to the recorded speech for individual words. The *Speak Correct* system generates comprehensive feedback by highlighting the mispronounced phonemes and providing a description of the articularity features of the phonetic letters, to encourage the correct pronunciation.

To date, 20 participants have completed the testing process. They all used the same devices (laptop, headsets, etc.), and were required to go through 2 levels with 10-35 lists of words, and to complete all the lessons for each level. Two groups of learners participated in this phase (a Saudi group and Egyptian group). Most of the words were pronounced correctly, as shown in figure 9.

Figure 8 describes the pronunciation guide with visual feedback. The feedback includes suggestions and explanations about things the user could try if they are experiencing difficulty. The feedback tab features a picture of all the items the user has attempted. Finally, the evaluation offers a lesson summarization to the students for each level.



```
⊞ Level#1
⊟ Level#2
    ⊞ Lesson#1
    ⊞ Lesson#2
    ⊞ Lesson#3
    ⊞ Lesson#4
⊞ Level#3
⊞ Level#4
⊞ Level#5
⊟ Level#6
    ⊟ Lesson#1
        bed ==> Not yet
        best ==> Done
        bird ==> Not yet
        egg ==> Not yet
        end ==> Not yet
        head ==> Not yet
        ten ==> Not yet
    ⊟ Lesson#2
        bird ==> Not yet
        burst ==> Not yet
        earn ==> Not yet
```

Figure (8): The Levels and Associated Lessons in the Speak Correct System

5.4 Speak Correct Experimental Results

We checked the accuracy of the *Speak Correct* system using the Word Error Rate (WER), which is derived from Levenshte in distance [36]. The WER can be calculated according to the following equation: $WER = (P_{sub} + P_{del} + P_{ins}) / N$

Where P_{sub} represents the number of substitution phonemes;

P_{del} represents the number of deletion phonemes;

P_{ins} represents the number of insertion phonemes; and

N is the total number of errors.

Our experiment was designed to evaluate the proposed *Speak Correct* system based on a phonetically proposed corpus, regardless of the speaker's accent or gender. The calculated value of the WER showed satisfactory performance of the *Speak Correct* recognition system.

The evaluation technique in the *Speak* Correct system is based on system responses that measure the degree of performance accuracy. Analysis of mispronunciation during the evaluation process is vital due to the complexity of speech processes and existence of tunable thresholds and parameters. Our results illustrate that an Egyptian accent from a region in the middle of Egypt (Cairo) has a high rate of correct recognition compared to an Egyptian accent from northern Egypt (Alexandria). Most of the words pronounced by speakers from Cairo were correctly recognized by the *Speak Correct* system.

In summary, we found that recognition varied based on regional accent. Table 2 and table 3 illustrate the distribution of speakers (Training and Testing) with respect to each region.

We found significant regional differences in speech recognition accuracy in terms of *Speak Correct* features. In Cairo, the *Speak Correct* performance accuracy was very high (the English language is considered to be a very important language in this region, and is generally used at school). Alexandria, the second capital city (after Cairo) had a good recognition rate. The worst score was obtained for speakers in the Rabegh locality in Saudi Arabia.

Table 2: Speaking (for Training and Testing) by Regions

Region / Localities		Training		Testing	
		Male	Female	Male	Female
Egypt	Cairo	17	17	8	8
	Alexandria	18	18	9	9
Saudi Arabia	Jeddah	10	10	3	3
	Rabegh	8	7	4	4

Table 3: Word Error Rate (WER) Relative to Different Regions

Region / Locality		Testing	
		Male	Female
Egypt	Cairo	10 %	11 %
	Alexandria	11 %	12 %
Saudi Arabia	Jeddah	16 %	17 %
	Rabegh	19 %	20 %

Acknowledgement

The teamwork of the ***Speak Correct*** project was funded as part of the strategic technology project (10-INF-1406-03) held at the King Abdulaziz University (KAU). Also, the authors wish to thank King Abduaziz City for Science and Technology (KACST) for funding, which was received through grant number 10-INF-1406-03. This financial support during the research period is gratefully acknowledged.

Conclusion

This paper focused on developing a *Speak Correct* phonetic editor system to be usedin adding new teaching levels and related lessons for speech correction for non-native English speakers. The proposed editor includes an interactive hint system to motivate users to improve their language skills.

Our experiment was designed to evaluate the *phonetic editor of the Speak Correct* system based on a phonetically trained corpus, regardless of the speaker's accent or gender. Our findings indicate thatthe proposed phonetic editor of the *Speak Correct* performs satisfactorily as a speech recognition system.

References

[1]. Macherey K., Bender O., Ney H., (2009). Applications of Statistical Machine Translation Approaches to Spoken Language Understanding, Audio, Speech, and Language Processing, IEEE Transactions on Volume: 17 , Issue: 4.

[2]. De Mori R., Bechet F., Hakkani-Tur D., McTear M., Riccardi G.,Tur G., (2008). Spoken language understanding, Signal Processing Magazine, IEEE Volume: 25 , Issue: 3.

[3]. Dinarell M., Stepanov E. A., Varges S., Riccardi G., (2010). The LUNA Spoken Dialogue System: Beyond utterance classification, Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference.

[4]. Camelin N., Bechet F., Damnati G., De Mori R., (2010). Detection and Interpretation of Opinion Expressions in Spoken Surveys, Audio, Speech, and Language Processing, IEEE Transactions on Volume: 18 , Issue: 2.

[5]. Laskowski K., Shriberg E., (2010). Comparing the contributions of context and prosody in text-independent dialog act recognition. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference.

[6]. Cantrell R., Scheutz M., Schermerhorn P., Xuan Wu, (2010). Robust Spoken Instruction Understanding for HRI. Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference.

[7]. Song Y., Wang Y., Ju Y., Seltzer M., Tashev I., and Acero A., (2009). Voice search of Structured Media Data, Acoustics, Speech and Signal Processing, 2009. ICASSP, IEEE International Conference.

[8]. Heracleous P.,Aboutabit N., Beautemps D., (2009). HMM-based vowel and consonant automatic recognition in Cued Speech for French, Virtual Environments, Human-Computer Interfaces and Measurements Systems. VECIMS '09. IEEE International Conference.

[9]. Kain A., Santen J. V., (2009). Using Speech Transformation to Increase Speech Intelligibility for the Hearing- and Speaking-Impaired, Acoustics, Speech and Signal Processing. ICASSP 2009. IEEE International Conference.

[10]. Tsubota Y., Kawahara T., Dantsuji M., (2002). Recognition and Verification of English by Japanese Students for Computer Assisted Language Learning System. In Proc. ICSLP, pp. 1205-1208.

[11]. Wohlmayr M., Stark M., and Pernkopf F., *(2011).* A Probabilistic Interaction Model for Multi-pitch Tracking With Factorial Hidden Markov Models. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 19, No. 4, May 2011.

[12]. Tsubota Y., Kawahara T., Dantsuji M., (2004). Practical Use of English Pronunciation System for Japanese Students in the CALL Classroom. In Proc. ICSLP, pp. 1689-1692.

[13]. Katsutoshi I., Komatani M., Ogata K., Okuno T., Hiroshi G., (2011). Simultaneous processing of sound source separation and musical instrument identification using Bayesian spectral modeling. (ICASSP), IEEE

International Conference on Acoustics, Speech and Signal Processing, 22-27 May 2011.

[14]. Huijbregts M., LeeuwenD. M., (2011). Unsupervised Acoustic Sub-word Unit Detection for Query-by-example Spoken Term Detection. IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP), on 22-27 May 2011, page(s): 4436 – 4439, Prague, Czech Republic.

[15]. Wang H., Christopher J. and Kawahara T., (2009). Computer Assisted Language Learning System based on Dynamic Question Generation and Error Prediction for Automatic Speech Recognition, Speech Communication 51, (2009), pp. 995 – 1005.

[16]. Yang H., Pu Y., Wei H., and Zhao Z., (2004). An Acoustic-Phonetic Analysis of Large Vocabulary Continuous Mandarin Speech Recognition for Non-native Speakers, Chinese Spoken Language Processing, International Symposium on 2004, pp. 241 – 244.

[17]. Abdou S., Rashwan M., Al-Barhamtoshy H., Jambi K., Al-Jedaibi W.. (2014) "Speak Correct: A Computer Aided Pronunciation Training System for Native Arabic Learners of English", Life Science Journal, 11 (X),), http://www.lifesciencesite.com

[18]. Fujii K., Yoshioka T., Yamasaki K., Muneyasu M. and Morimoto M., (2011).A Double Talk Control Method Improving Estimation Speed by Adjusting Required Error Level, Workshop on Hands-free Speech Communication and Microphone Arrays, IEEE May 30 - June 1, 2011.

[19]. Kac J. and Rozinaj G., (2009). *Adding Voicing Features Into Speech Recognition Based on HMM in Slovak,* IEEE Conference, Systems, Signals and Image Processing, IWSSIP 2009,16th International Conference.

[20]. Abdou S., Rashwan M., Al-Barhamtoshy H., Jambi K. and Al-Jedaibi W., (2012). *Enhancing the Confidence Measure for an Arabic Pronunciation Verification System.* Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012, KTH, Stockholm, Sweden.

[21]. Jurafsky D. and Martin J. H., University of Colorado, Boulder, (2008). Speech and Natural Language Processing. 2nd Edition, Prentice Hall.

[22]. Zhou Y. and Shang Li, (2012). Speaker Recognition Based on Principal Component Analysis and Probabilistic Neural Network, Lecture Notes in Computer Science, 2012, Volume 6839, Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, pp 708-715.

[23]. Herbig T., Gerl F., and Minker W., (2012). Self-learning speaker identification for enhanced speech recognition. Computer Speech & Language, Vol 26, Issue 3, June 2012, pp. 210–227.

[24]. Witt S. M., (2012). Automatic Error Detection in Pronunciation Training: Where we are and where we need to go. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012 KTH, Stockholm, Sweden, (IS ADEPT, Stockholm, Sweden, June 6-8 2012).

[25]. Pellom B., (2012). Rosetta Stone ReFLEX: Toward Improving English Conversational Fluency in Asia. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012 KTH, Stockholm, Sweden, (IS ADEPT, Stockholm, Sweden, June 6-8 2012).

[26]. Smith B. (2011). Arabic Speakers: Learner English, Cambridge Handbooks for Language Teachers, 2nd Edition, Series Editor Scott Thornbury.

[27]. Zhu, J., Wang, H., Hovy, E. H. (2010). Confidence-based Stopping Criteria for Active Learning for Data Annotation. ACM Transactions on Speech and Language Processing, Vol. 6, No. 3, Article 3, Publication date: April 2010.

[28]. Zhu, J., Wang, H., and Hovy, E. H. (2008). Multi-Criteria-Based strategy to stop active learning for data annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics.* 1129–1136.

[29]. Knight K. and May J., (2009).Handbook of Weighted Automata, Edited by Manfred Droste, Werner Kuich, Heiko Vogler, Springer. Chapter 14: Applications of Weighted Automata in Natural Language Processing.

[30]. Brants T., Popat A. C., Xu P., Och F. J., and Dean J., (2007). Large language models in machine translation. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 858–867, Prague, June 2007.

[31]. Galley M., Hopkins M., Knight K., and Marcu D., (2004).What's in a Translation Rule? In HLT-NAACL Proceedings, 2004.

[32]. Gildea D., (2003). Loosely tree-based alignment for machine translation. In ACL Proceedings, Sapporo, Japan, 2003.

[33]. Graehl J. and Knight K., (2004). Training Tree Transducers. In HLT-NAACL Proceedings, 2004.

[34]. Knight K. and Graehl J., (2005). An Overview of Probabilistic Tree Transducers for Natural Language Processing. In CICLing Proceedings, 2005.

[35]. Kumar S. and Byrne W., (2003). A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In HLT-NAACL Proceedings, 2003.

[36]. May J. and Knight K., (2006). A Better n-Best List: Practical Determinization of Weighted Finite Tree Automata. In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pages 351–358, New York City, USA, June 2006. Association for Computational Linguistics.

[37]. May J. and Knight K., (2006). Tiburon: A Weighted Tree Automata Toolkit. In Oscar H. Ibarra and Hsu-Chun Yen, editors, Proceedings of the 11th International Conference of Implementation and Application of Automata, CIAA 2006, volume 4094 of Lecture Notes in Computer Science, pages 102–113, Taipei, Taiwan, August 2006. Springer.

[38]. Zuo X., Sumii T., Iwahashi N., Nakano M., Funakoshi K., Oka N., (2013). Correcting Phoneme Recognition Errors in Learning Word Pronunciation through Speech Interaction, Speech Communication, 55, pp190-203, 2013.

[39]. Mohamed A., Dahl G., Hinton G., (2009). Deep Belief Networks for Phone Recognition. In: Proceedings of the 22nd Neural Information Processing Systems Conference (NIPS) Workshop on Deep Learning for Speech Recognition, pp. 1-9, 2009.

[40]. Leitner C., Schickbichler M., Petrik S., (2010). Example-based Automatic Phonetic Transcription. In Proceedings of Seventh Conference on International Language Resources and Evaluation (LREC -2010) pp. 3278-3284.

[41]. Nakagawa S., (2006). Spontaneous Speech Recognition: Its Challenge and Limit. In: Proceedings of the IEICE General Conference, Japanese Edition, pp. 13-14, 2006.

[42]. Soong K., Lo K., Nakamura S. (2004). Generalized Word Posterior Probability (GWPP) for Measure Reliability of Recognized Words. In: Proceedings of the Special Workshop in Maui (SWIM), 2004.

[43]. Nakamura S., Markov K., Nakaiwa H., Kikui G., Kawai H., Jitsuhiro T., Zhang J., Yamamoto H., Sumita E., Yamamoto S., (2006). The ATR Multilingual Speech-To-Speech Translation System. IEEE Trans. Audio Speech Lang. Process. 14, 365-376, 2006.

[44]. Droua-Hamdani G., Sellouani A., and Boudraa M., (2013). Effect of Characteristics of Speakers on MSA ASR Performance, IEEE 2013.

[45]. Al-Barhamtoshy H., Abdou S., Jambi K., (2014) "*Pronunciation Evaluation Model for None Native English Speakers*", Life Science Journal, 11 (9), http://www.lifesciencesite.com

6/2/2014