Design and Development of an

Automatic Speech Recognition System for Urdu

by

Agha Ali Raza

A thesis presented to

FAST-National University of Computer and Emerging Sciences

In partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

NUCES-FAST, Lahore, Pakistan, July 2009

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled *Design and Development of an Automatic Speech Recognition System for Urdu* by Agha Ali Raza in partial fulfillment of the requirements for the degree of Master of Science (Computer Science).

____

Head
(Department of Computer Science)

Committee Members:

Thesis Supervisor

____

Dr. Sarmad Hussain
Professor
FAST - National University

Thesis Co-Supervisor

____

Mr. Awais Athar
Lecturer
FAST - National University

Dated: July 2009

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my supervisors.

I understand that my thesis may be made electronically available to the public.

Signatures **Date**

Abstract

Easy access to information is one of the key factors contributing to the economic and social growth of societies in the modern era. Most of the available means of information access like print media are useful only for the literate members of the society. Other modes like television and radio are non-interactive and computers, although being interactive are not suitable for the major portion of the society that is either not familiar with its interface and usage or does not have access to it at all. One solution to this problem can be a telephone based speech interface for human-computer interaction. As an Automatic Speech Recognition System (ASR) is a key component of such an interface, the prime impediment to the achievement of this goal in Pakistan is the lack of research and local language resources for Urdu that are required for the development of an Urdu ASR. On an abstract level a speaker independent automatic, *continuous*[1] and *spontaneous*[2] speech recognition system for local languages and its further adaptation to telephone based interface is required as a first step towards achieving this goal.

CRULP is currently working on a project entitled "Telephone-based Speech Interfaces for Access to Information by Non-literate Users" in collaboration with the Carnegie Mellon University. The goal of this project is to investigate the use of speech interfaces in a field-deployed system by providing easy access to medical information to lady health workers in Pakistan. This will be achieved by developing a telephone based dialogue system consisting of an Urdu Speech Recognition system and a Text to Speech system that can interact with the health workers to answer their queries.

One of the main components of this system is a core Urdu Speech Recognition system that can be trained with field specific data at a later stage. I have taken that to be the goal of my MS Thesis. This thesis presents the design and development of a medium vocabulary[3] ASR system for spontaneous Urdu speech. The targeted domain is the accent spoken by literate speakers in the suburban areas of Lahore. The system is trained for continuous read as well as spontaneous speech and has been adapted to home and office level back ground noise. In this initial phase the system is speaker specific. This involves the design and development of speech corpora for read and spontaneous Urdu speech, and ASR system training for Urdu. CMU Sphinx system has been used as the primary training and recognition engine.

[1] Where words are not necessarily separated by silence i.e. a lexicon entry does not necessarily map onto an individual utterance

[2] As naturally spoken by speakers in everyday life where the processes of planning and speaking go side by side in contrast to reading out prepared data

[3] 10,000 words, approximately

Acknowledgements

Five years ago, a mere spark of curiosity led me to register in the course of *Phonetics and Phonology*. Little I anticipated that Dr. Sarmad will leave me with a burning flame of fascination for this field and it will become the nucleus of my graduate study. I am greatly indebted to my thesis supervisor, Dr. Sarmad Hussain, for his help and continuous support while I desperately kicked around in the unknown waters of Speech Processing and finally learned to enjoy the quest for knowledge. I am also very thankful to Sir Belal Mohammad Hashmi for his role resembling an *oracle* in my thesis. Towards him, I used to look whenever I lost my way in the tortuous alleys of computer hardware. And many thanks to my thesis co-supervisor and colleague Mr. Awais Athar for keeping me on my toes throughout my thesis research through continual interrogation and constant criticism and for nudging me back to work whenever I slacked.

I wish to express my gratitude towards the team of researchers of the *Center for Research in Urdu Language Processing* (CRULP), working on the project entitled *Telephone-based Speech Interfaces for Access to Information by Non-literate Users*, for their help and support, as many parts of my thesis overlapped with that project. I would like to thank Ms. Huda Sarfraz and our brilliant linguists Mr. Inam Ullah and Mr. Zahid Sarfraz for their eager support throughout the thesis. I found them to be the most difficult to convince persons in matters of Phonetics (with the exception of Dr. Sarmad), and this was precisely what used to trigger my search for answers based on solid arguments. I wish to thank my batch fellows Mr. Umer Khalid, Mr. Zeeshan Latif, Mr. Ahmad Muaz, Mr Imtiaz Ahmad and my students Mr. Ameer Sheikh, Mr. Atif, Mr. Sayyam Mehmood, Mr. Talha Ahmad, Ms. Maria, Ms. Nida Fatima and Ms. Jweria Ghazanfar for gladly volunteering for the speech recordings and for their patience in uttering phrases and sounds which make the human tongue realize that there are yet unexplored regions in the oral cavity.

Finally I am greatly thankful to the CRULP for generously providing me with valuable linguistic resources and an open access to the technical expertise of its researchers and linguists. I am grateful to *FAST-National University of Computer and Emerging Sciences* for supporting my thesis.

Above all I want to thank my caring mother without whom I would have given up a long way back. She is still the only person who holds my finger and compels me to jump over yet another hurdle. All the praise and all the glory rightly belongs to my Almighty Allah, my dear and powerful companion whose hand supports me when my frail knees buckle under the weight of my responsibilities.

I dedicate this work to my Lord Allah and my parents.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

The aim of *Automatic Speech Recognition* research is to develop computational techniques to convert acoustic speech signals into strings of words. The problem of general speech recognition has not been solved yet for any language. This means that there exists no such versatile speech recognition system which can recognize the words spoken by any person, of any gender, in any accent and at any rate of speech in any environment. Therefore the task of speech recognition starts by making simplifying assumptions which manifest as the constraints on the system. In other words, all the open ended variables are constrained to certain practical limits which render the problem more feasible to tackle.

There are many areas where Automatic Speech Recognition (ASR) systems can play a pivotal role in facilitating the daily activities, and where the current levels of accuracy that these systems have attained can prove useful. One of these areas is speech based *human-computer interaction* (HCI). This line of research promises to be of significant advantages in areas where keyboards may not be appropriate and natural language communication is desired. This includes control applications where hands and eyes may be busy at the same time and speech becomes a good means of issuing the commands. In addition to this, such systems can be of immense use for people with vision related disabilities, lack of motor control, crippled hands etc. In the under-developed countries where literacy rate is poor, this can provide a mechanism of information access to people who are unable to read and write as well as people who may be literate but not qualified in computing skills. Speech based HCI ideally brings computers within reach of anyone who can speak and listen. The major hurdle however, is the lack of the resources required to develop these systems for the native languages of underdeveloped countries. Another major area of application for ASR systems is telephony where such a system can provide recognition for digits or simple commands in the form of yes/no questions. These two application areas can be combined into one by allowing a complete telephone based HCI for computers, which is the goal of the *Telephone-based Speech Interfaces for Access to Information by Non-literate Users*, project currently being done by a joint effort of the *Center for Research in*

Urdu Language Processing (CRULP) and *Carnegie Mellon University* (CMU) and is also the prime motivation behind this thesis. Other useful applications include ASR based dictation systems and speech transcription systems for meetings and conferences. Control systems for mobile phones, automobiles and aircrafts are rapidly becoming feasible and are commercially available.

As discussed earlier, the problem of general speech recognition is as yet not practically solved; therefore the domain is restricted by using simplifying assumptions. These assumptions are dependent upon the application areas of the ASRs as they are dependent upon the practicality of the resulting system requirements. The task is to take the variables involved in the ASR system design and restrict them one by one to feasible constraints. Following are some of these variables and sample parameters which can be used to restrict them:

- **Language:** The systems are generally language specific. Therefore currently the ASR systems are trained for data of a specific target language.
- **Speaker Class:** The speaker class may be defined in a variety of ways depending on the acoustic and phonetic variations represented by these classes.
    - **Literacy:** There may be a lot of variation in the pronunciation and articulation methods of people with different literary backgrounds, and it may be advisable to predefine the target literacy level [1].
    - **Area of residence:** This may in some cases also capture the literacy class as well as the dialect. The area of residence or the area where a speaker has spent most of his/her life may represent his/her phonetic classification.
    - **Accents and Dialects:** Languages often have numerous dialects which lead to a great variation of pronunciation even within a single language. Therefore the target dialect (or the set of target dialects) has to be defined before the ASR system design. Moreover, unfamiliar accents i.e. the accents which are different from the ones on which a system has been trained pose a challenge to recognition. For example, a person from Sindh or Northern Areas of Pakistan using an ASR trained with Urdu in the Lahore suburban accent may not get good recognition results. Therefore, separately trained systems may be required for

recognizing different accents. Compatibility of various dialects and accents may have to be phonetically verified before such decision making.

  - **Age:** The acoustic properties of speech like pitch, formants etc. are stable in ages between 20-45 years. However, children and old aged individuals may represent variable acoustic properties that may make the system commit more mistakes. Therefore, different systems may be required for children, middle aged and old aged individuals.
  - **Gender:** The male and female voice is different in many basic characteristics like pitch and format placement. Therefore, a system targeted towards male and female speakers both, will require balanced amounts of training data from both the genders. It may even require separately trained systems for the two genders working with a front end of gender recognition for better results.

- **Vocabulary size:** This represents the number of distinct words the target system is supposed to recognize. Systems may vary from *small vocabulary* systems with a vocabulary of a few words e.g. a yes/no system with a vocabulary size of two or a digit recognition system with a vocabulary size of tens, to *medium vocabulary* and *large vocabulary* systems with vocabularies of sizes below 20,000 or above 20,000 to 60,000 words respectively. The systems may be trained in terms of words, syllable, phones or combination of phones (like diphones or triphones). The meaning of vocabulary size will be different for each of these systems. A system trained on all phones of a language may be able to recognize all (or most of) the words in that language although the actual number of words actually in the phonetic lexicon are much less.
- **Degree of Fluency:** The degree of fluency represents the rate of speech and in turn represents the chances of wrong or incomplete pronunciations.
  - **Isolated:** By far this is the simplest constraint. The words are required to be preceded and followed by pauses (i.e. silence). Digit recognition systems and yes/no systems (command driven systems) often fall into this category. The words are mostly carefully uttered and are hence are by and large complete and error free.

- **Continuous:** These are the systems where words may run into each other and hence segmentation of speech into sound units becomes a challenge. These may further be divided into many sub categories. For example, there are systems in which the pace of speech is controlled, like dictation systems where the speech is carefully articulated at a constant pace. Furthermore, there are systems in which a person knowingly issues commands to a computer system and hence is often careful in articulation and pronunciation (this may lead to unnatural pronunciations as well). In general we can make the following distinctions:
    - **Read Speech:** Continuous speech read from some text is generally characterized with careful pronunciation and a controlled and consistent pace of speaking [1].
    - **Spontaneous:** This is the most difficult to model variation, for example when a human talks to another human as in meetings or conferences, or over telephones etc. [1] The rate of speaking varies greatly as speech planning and delivery go side-by-side, the words may be mispronounced and/or incomplete as the humans rely on non-verbal communication, experience, context etc. to understand and rectify speech errors. The spontaneous speech also possesses many subcategories like for example the spontaneous speech uttered in broadcast news or television or radio shows may be more carefully articulated compared to the speech in everyday human conversation where there might be too many disfluencies like repetitions, incomplete words and pauses etc.

- **Environment:** Speech environment has to be predetermined as it shows a wide variation from controlled noise studio environments, to low ambient noise house and office environments leading to the extremely noisy automobile, air craft environments and busy street environments.
- **Channel:** The recording channel may be a simple microphone attached to digital speech acquisition hardware, a telephone based system employing VoIP techniques or a mobile phone channel characterized by packet losses etc. Every channel introduces its own characteristics into the speech like frequency limits (e.g. a 16000 or 44100 sample per

second microphone based speech vs. the 4000 Hz, 8000 sample per second speech for a telephony system). Besides the characteristic channel noise due to any of numerous factors like channel properties (which may remain consistent) to variable factors like vicinity of electronic equipment (which varies greatly) are some of the salient features of speech environments.

- **Online vs. Offline Systems:** The efficiency requirements on a live or online system which works under some real-time constraints are much different from an offline system which transcribes recorded audio (where even a 10 times (10x) real time limit may be tolerated). Examples of the former are interactive systems like the speech based control systems for devices while transcription systems for meetings and discussions may be implemented as offline systems.

1.1 Speech Recognition Architecture

The main task of a Speech Recognition system is to take an acoustic signal as input and produce a string of words as output. In HMM based speech recognition systems this is modeled by using a *noisy channel model [1].* The idea is to assume that the acoustic signal is a noisy version of the string of words. Hence, in essence we model the acoustic system, acoustic variation as well as the environment and channel noise all as a channel which adds noise to the string of words. In order to extract the original string of words from this noisy string, we need to know how the channel modifies the string. So if we have a model for the channel, we can pass every sentence in the language through that model and *compare* it with the noisy string of words to find the original string of words.

Let us intuitively review the architecture of a speech recognition system. We train the system by giving it recorded speech data and its transcribed form (the original string of words). With this data, we try to find a pattern for the noisy versions of all the basic sound units in the language (if we are making a phone based system) or the noisy versions of all the words in the language (if we are making a word based system). This is called the *training phase* and it provides us with the *Acoustic Model.* To this information we add the language dependent information i.e. the probability of words occurring in different contexts, in what is called the

Language Model. Together the language model and the acoustic model help us convert the input acoustic signal to a string of words. This later process is called the *decoding phase*.

So, a speech recognition system asks the following question [1]:

Given some acoustic observation O, what is the most likely sentence out of all the sentences in the language?

Where, O is a sequence of individual observations obtained by segmenting the input wave form into representative chunks of particular durations:

$$O = o_1, o_2, o_3, \ldots, o_t \tag{1.1}$$

Let us define a sentence W as a string of words w:

$$W = w_1, w_2, w_3, \ldots, w_n \tag{1.2}$$

The words defined here are based upon orthography, i.e. نو (new) and نو (nine) will be considered as same while, لڑکا (boy) and لڑکے (boys) will be considered as different words. So the probabilistic interpretation of our question becomes:

$$W' = \underset{W \in L}{\operatorname{argmax}} P(W|O) \tag{1.3}$$

Where W' is the required string of words (sentence) and L represents the set of all sentences in the language. As there is no direct way of calculating this, we may simplify it by using the Bayes' Rule, defined as:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{1.4}$$

So applying it on Equation (1.3) we get:

$$W' = \underset{W \in L}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} \tag{1.5}$$

The probability in the denominator is not a simple one to calculate. However, as we are only interested in the maximum value we can remove it as the common denominator to give:

$$W' = \underset{W \in L}{\operatorname{argmax}} P(O|W)P(W) \qquad \textbf{(1.6)}$$

So, we are saying that the most probable sentence W given some observation sequence O can be computed by taking the product of two probabilities: the $P(W)$ or the *prior probability* comes from the *language model*, while the $P(O|W)$ or the *observation likelihood* is computed by the *acoustic model*.

So the Automatic Speech Recognition System is composed of a trainer which trains the *P(O/W)* and *P(W)* using a particular data set. The trained system can then be used to decode (recognize) input speech *O,* to give a string of words *W* as output. Figure 1 depicts this procedure.

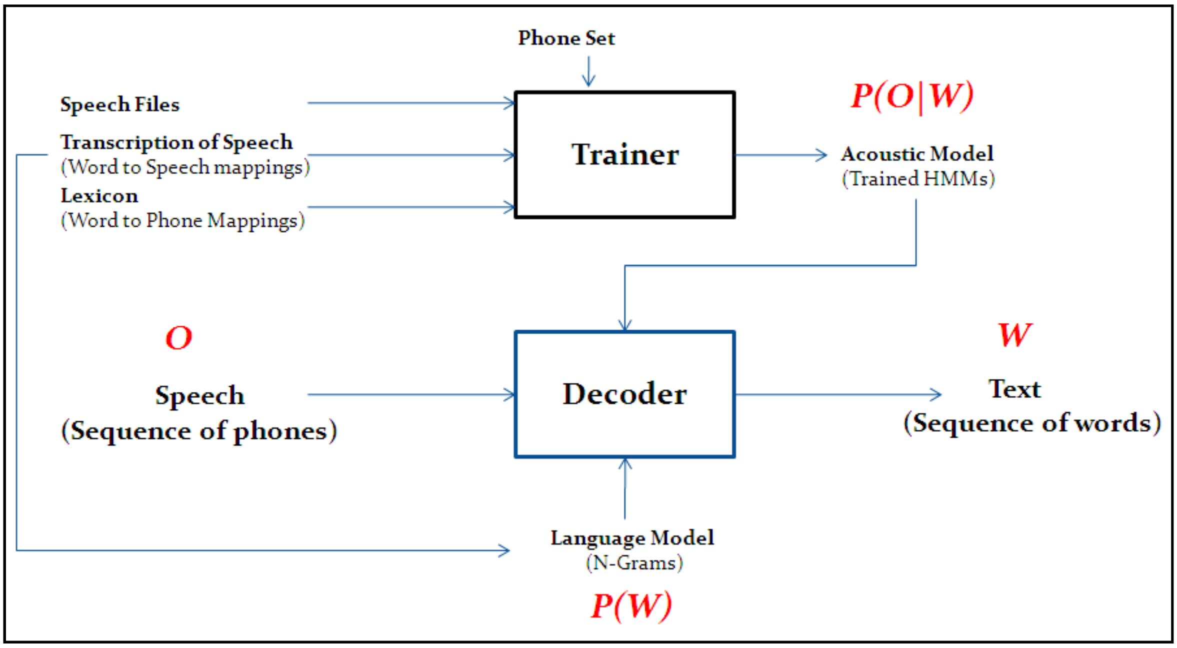


Figure 1 - The Automatic Speech Recognition System Architecture

In the remaining discussion we shall see how we can estimate these two probabilities.

1.2 The Acoustic Model

The acoustic model establishes a mapping between phonemes and their possible acoustic manifestations, i.e. the phones. So given an observation i.e. a slice from a digitized speech waveform, we have to designate the most closely matching phoneme. Looking at the high level design, the acoustic model training process takes three main types of data as input. A set of recorded speech files, a text file containing parallel transcription of these speech files and a phonetic dictionary. The phonetic distionary maps all the words in the transcription file to their constituent phonemes (these can be as finely grained as phones or as coarse grained as words). The training process maps all the occurrences of phonemes onto the acoustic set of phones.

Let us assume there were only two phonemes in the corpus, /b/ (voiced, bilabial stop) and /k/ (unvoiced, velar stop) (as shown in Figure 2). Now by the very nature of speech and human articulatory system, even a single phoneme cannot be uttered in *exactly the same way* twice (spectrally). However, the acoustic properties of an utterance of /b/ will be more similar to other utterances of /b/ than to those of /k/. In this way, we can make clusters of the acoustic realizations for all the required phonemes. In the decode phase when a new observation is presented to the trained system, it is matched with all the available clusters using the same matching criteria as was used in training and designated to the one which it most closely resembles. These criteria will be defined later.

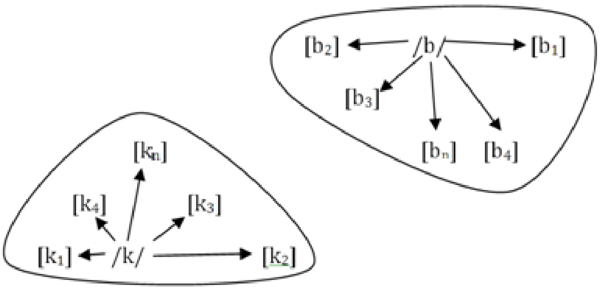


Figure 2 - Phone Clusters

This simple description of speech recognition systems needs more details. For example, phones, the acoustic manifestations of phonemes, are not independent temporal entities in an utterance. They are affected by the phones following and preceding them. Hence some weight must be given to the acoustic context. Then there is the question "*Are all phones as easy to cluster as our example of [b] and [k]?*" The answer is, "*No*". There are many phones which closely resemble others (e.g. [o] and [ɔ]) and their clusters are not so trivial to classify. Hence, acoustic context, possible phone combinations and history will play an important role in the recognition. Finally, the acoustic properties do not remain constant even in a single phone, so for classification phones are broken down to smaller units, mostly consisting of three stages, the beginning, middle and end of phone.

Human speech is produced as air from the lungs rushes out through the larynx producing vibrations in the vocal folds and/or noise in any regions of the oral or nasal cavity [2] (see Figure 3). This sound is modified by the change in shape and size of the tracts as the oral and/or nasal tract is completely or partially obstructed. A typical open vocalic sound like [∂] is characterized by vocal-fold vibrations producing a periodic waveform, which produces stationary waves in the (open) oral tract. Some of the frequencies in the glottal waveform get suppressed while others are reinforced in the oral tube. The reinforced portions exhibit a periodic pattern and show peaks at roughly 500 Hz, 1500 Hz, 2500 Hz and so on in case of a average adult (with a 17.5 cm long oral tube like an average American ([3], [4])). These resonances are called formants which hold a key significance in characterizing vocalic sounds. Nasal vocalic sounds are produced when the velo-pharyngeal port is open and the air partially escapes through the nose, producing a dampening effect on the formants and they become wider, shifted and lower in amplitude. Thus the oral and/or nasal tracts act as a filter which modify the source waveform produced by the voice box (trachea). The consonants are produced by partial or complete blockage of air flow in different portions of the oral or nasal tract. This results in an explosive or noisy egress of air from the mouth or nose resulting in the consonantal sounds.

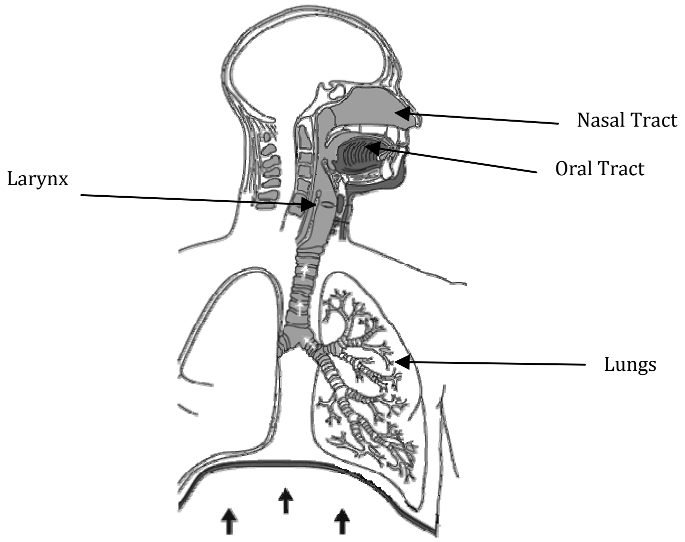


Figure 3 - The Human Articulatory System [5]

The first step in speech recognition is to decide upon the phone representation. After recording, the speech is available as a digitized time-varying waveform. We have to split this waveform, obtained for the complete utterance, into segments. Then we need to represent each segment using some technique which would help in the process of identifying it as a separate phone. In other words we want to bring out those features which make any phone different from other phones.

The splitting of the utterance into segments of around 10ms duration is accomplished using windowing [1] (Figure 4 (part-2)). Using rectangular window introduces sharp discontinuities at the edges. These discontinuities appear as impulses and introduce white noise in the frequency response; therefore a more tapering window function is preferred, like a Hamming window.

$$Rectangular: w[n] = \begin{cases} 1, & 0 \le n \le L-1 \\ 0, & otherwise \end{cases} \qquad \textbf{(1.7) [6]}$$

$$Hamming: w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L}\right), & 0 \le n \le L-1 \\ 0, & otherwise \end{cases} \qquad \textbf{(1.8) [6]}$$

After the signal has been split to achieve a sequence of windowed audio segments, the next step is to bring it into frequency domain. This is accomplished using the Discrete Fourier Transform, Figure 4 (part-3) or its more efficient version, the Fast Fourier Transform. The formula for the DFT is shown below:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi}{N}kn} \qquad \textbf{(1.9)}$$

This frequency domain signal can be used as a representative of the phones; however, the problem is that many other kinds of information which is not required in speech recognition are mixed together with the useful information in these signals. For example, the speaker-related information (evident from the pitch and some higher formants), information about intonation and stress etc. We are interested in separating out the useful information from the rest of the signal.

As the glottal waveform is characterized by a -12db/decade tilt [2], which is compensated by the radiation filter as sound leaves the mouth to a final -6db/decade tilt, we need to reinforce this wave before doing any further processing otherwise the higher formants will not be prominent. This is done using a pre-emphasis filter which tends to raise the higher frequencies, producing a more level spectrum Figure 4 (part-1). The next step is to adjust the energy distribution in different frequency bands to match the human perceptual response. The sensitivity of the Human ear increases from 20 Hz to about 1000 Hz and then begins to fall logarithmically [1]. The mapping is done to imitate that response by using the MEL (melody) scale Figure 4 (part-4) frequency mapping as given below in equation 1.10:

$$mel(f) = 1127 \ln\left(1 + \frac{f}{700}\right) \qquad \textbf{(1.10)}$$

Next we need to separate the glottal information from the vocal/nasal tract response. Many techniques can be used to accomplish this however, in speech recognition Cepstrum Analysis ([1], [6]) has been found to be of more use. The main idea of the Cepstum is to treat the frequency domain speech signal as a simple time domain waveform. As can be seen in Figure 4 (part-3) the spectrum of the speech wave form contains some rapidly changing components which *ride* on slowly undulating peaks. The rapidly varying components are the harmonics of the pitch or the fundamental frequency i.e. the *source* (in the terminology of the *source-filter theory* [6]). The slowly varying components are the formants, i.e. the *filter* response. The Cepstrum Analysis is a technique which suggests to treat this spectrum as just an ordinary waveform and to separate out the high *frequency* and low *frequency* components that it contains. The problem is that from *source-filter theory* we know that these source and filter responses do not exist as a sum, which could have been easily filtered in frequency domain using high pass and low pass filters, but as a product. Therefore, we need to use the logarithm to convert the product into sum first. Next we take a DFT of the spectrum. Taking the DFT of a spectrum should return us to Time-Domain, but the log prevents this to be a time domain signal. Instead we are in an inverse frequency domain, or *quefrency* domain [6]. This plot is called a *Cepstrum*. As can be clearly seen in Figure 4 (part-5), the source and the filter response are now well separated. And can easily be separated by filtering (*liftering* in Cepstral terminology). This gives us the log domain source and the log domain filter response. This can be converted back into frequency or time domain by taking inverse DFT and antilog. Mathematically the Cepstrum for a windowed frame of speech can be represented as:

$$c[n] = \sum_{n=0}^{N-1} \log \left(\left| \sum_{\mathrm{n}=0}^{\mathrm{N}-1} \mathrm{x[n]e}^{-\frac{\mathrm{j}2\pi}{\mathrm{N}}\mathrm{kn}} \right| \right) \mathrm{e}^{\frac{\mathrm{j}2\pi}{\mathrm{N}}\mathrm{kn}} \qquad \textbf{(1.11) [1]}$$

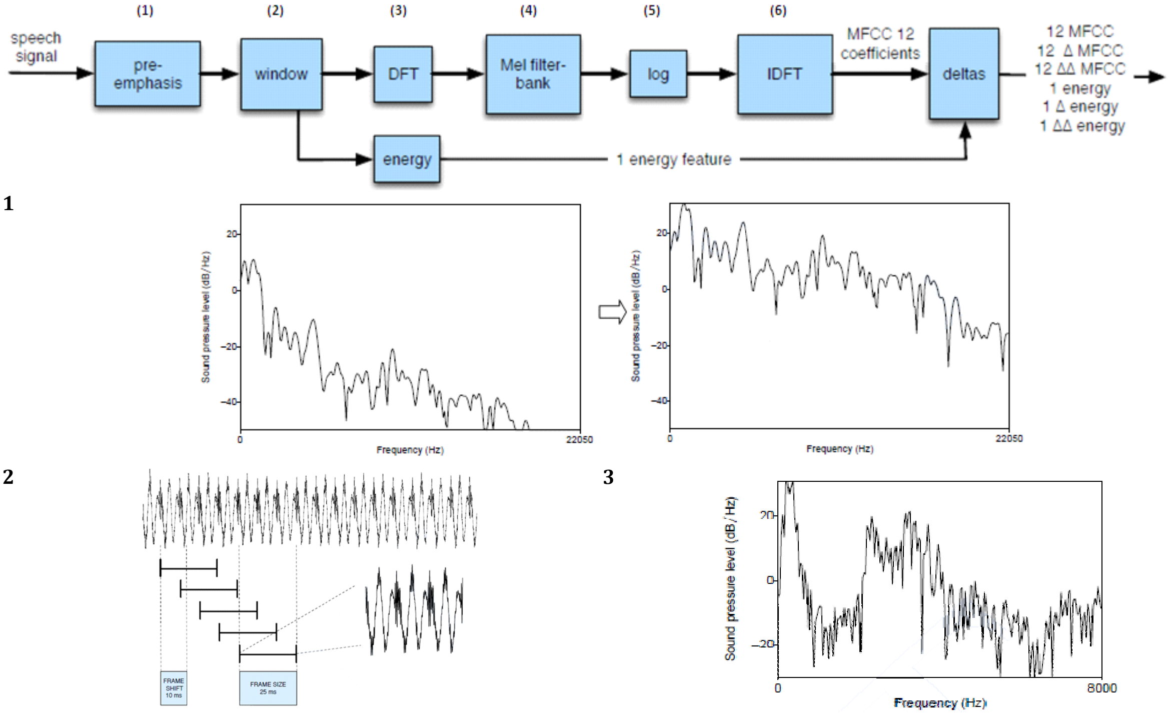
We are generally more interested in the first 12 cepstral coefficients that give the formant information required for recognition. In addition we also want the information regarding signal energy to discern between voiced and voiceless and vocalic versus consonantal phones. The energy in a frame is the sum over time of the power of the samples in the frame, thus for a signal *x* in a window from time sample t_1 to sample t_2, the energy is given as [6]:

$$Energy = \sum_{t=t_1}^{t_2} x^2[t] \qquad \textbf{(1.12)}$$

These 13 parameters form the basic set representing a phone. However, more information is required to capture the rising and falling trends of the formants as these are the characteristics of the places of articulation. Therefore, 13 deltas representing the rate of change of the 13 values (the velocity) and another 13 valued vector representing the rate of the rate of change (the double deltas, or the acceleration) are also stored. This gives us a 39 dimensional vector, representing a spectral slice. The Cepstral features so extracted are called the *Mel Frequency Cepstral Coefficients* (MFCCs). The summary of the MFCC features is as follows [1]:

- 12 cepstral coefficients
- 12 delta cepstral coefficients
- 12 double delta cepstral coefficients
- 1 energy coefficient
- 1 delta energy coefficient
- 1 double delta energy coefficient

The complete procedure from speech signal to MFCCs is shown in Figure 4.



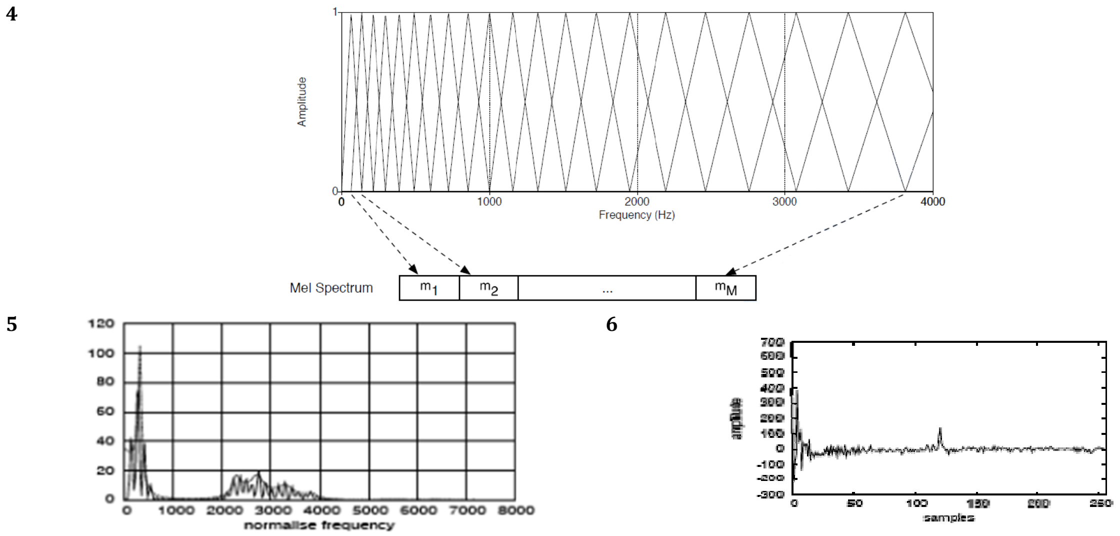


Figure 4 - Speech Signal to MFCCs [1], [6]

The acoustic model is implemented using Hidden Markov Models (HMMs) [1]. Although each state of an HMM can be mapped onto a single phone but as we just discussed it would be inappropriate as the spectral properties change drastically even within a single phone. A phone has spectrally three major and relatively stable portions: the beginning, which is a transition from the previous phone to the current one, the middle which depicts the actual properties of the current phone and the end, which is a transition from the current to the next phone. Therefore a phone is often modeled using five states of an HMM. The start and end are non-emitting and three central states model the three phases of the phone. The HMMs used for speech recognition fall into the category of Bakis Networks as in these there are no transitions to the previous states or jumps to skip states. The only two allowed transitions are to the next state or a self loop. The self loop allows modeling phones with varying lengths.

The complete HMM can be defined as:

- A set of states: $Q = q_1, q_2, \ldots, q_N$
- A transition probability matrix: $A = \alpha_{01}, \alpha_{02} \ldots \alpha_{n1} \ldots \alpha_{nn}$. Each α_{ij} represents the probability for each sub-phone of taking a self loop or going to the next sub-phone, such that $\alpha_{ii} + \alpha_{ij} = 1$ for all i
- A set of observations: $O = o_1, o_2 \ldots o_n$

- A set of observation likelihoods, also called emission probabilities: B = $b_j(o_t)$. Each expressing the probability of an observation o_t being generated from state *j*
- A special start and end state: q_0, q_{end}, that are not associated with observations

The state transition probability from state *i* to state *j* α_{ij} and the emission probability at state *j* $b_j(o_t)$ are trained using some Expectation Maximization algorithm (e.g. Baum-Welch [1]). The α_{ij} are trained using the information from the phonetic dictionary in which all the phonetic constituents of words are given. The $b_j(o_t)$, are modeled from the 39 dimensional MFCCs. The MFCCs are used to calculate, 39 dimensional multivariate Gaussian Probability Density Functions (PDFs). The PDFs are trained using Baum-Welch Algorithm. As it cannot be guaranteed that the cepstral coefficients will produce normal distributions in all cases so the PDFs are used to compute Gaussian Mixture Models (GMMs) [1].

Therefore the calculation of the value of $b_j(o_t)$ in a simplified model for a single cepstral feature, and assuming that each HMM state *j* has associated with it a mean μ_j and a variance σ^2_j, can be carried out as:

$$b_j(o_t) = \frac{1}{\sqrt{2\pi\sigma_j^2}} exp\left(-\frac{(o_t - \mu_j)^2}{2\sigma_j^2}\right) \qquad \textbf{(1.13) [1]}$$

In order to do the same with a *D*-dimensional feature vector (in our case *D*=39), given HMM state *j*, using a diagonal covariance multivariate Gaussian we use:

$$b_j(o_t) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{jd}^2}} exp\left(-\frac{1}{2}[\frac{\left(o_{td} - \mu_{jd}\right)^2}{\sigma_{jd}^2}]\right) \qquad \textbf{(1.14) [1]}$$

As discussed before that the assumption of normal distribution for all cepstral features is too hard a constraint. Therefore, the observation likely hood is often not estimated using a single multivariate Gaussian but a weighted mixture of multivariate Gaussians. The resulting model is called a Gaussian Mixture Model, which is trained using Baum-Welch to determine the observation likelihood $b_j(o_t)$.

One problem that may occur in the calculation of probabilities mentioned in this section is that of numeric underflow. Many small probabilities multiply to produce even smaller numbers. This problem can be solved by using *log probabilities* [1]. An additional benefit of working with probabilities in the logarithmic domain is that of enhanced computational speed. Instead of multiplying probabilities the log probabilities have to be added. And addition is a faster operation then multiplication. Thus 1.14 can be representated in log domain as:

$$logb_j(o_t) = -\frac{1}{2}\sum_{d=1}^{D}\left[\log(2\pi) + \sigma_{jd}^2 + \frac{\left(o_{td} - \mu_{jd}\right)^2}{\sigma_{jd}^2}\right] \quad \textbf{(1.15)}$$

1.3 **The Language Model**

The prior probability, *P(W)* is calculated using the *Language Model*. Generally trigram or even 4-gram based language models are used in modern speech recognition systems. For smaller systems that have to be deployed on embedded devices like mobiles phones etc. a bigram or even unigram model may be used to save space.

Briefly, an N-gram language model is constructed from a transcribed corpus by calculating the following probability:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \quad \textbf{(1.16)}$$

This is done for all word combinations present in the corpus. In order to keep this practical we approximate this probability by limiting *n* (where, *k=1…n)* to include previous 1 (bigram), 2(trigram) or 3(4-gram) words. We can even use simple occurrence probabilities of single words without considering the history, which is called a unigram model. For example, for a bigram language model the following probability is calculated for all the words in the corpus:

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1}) \quad \textbf{(1.17)}$$

The simplest way to calculate these probabilities is by using the Maximum Likelihood Estimation (MLE). This is done by using normalized (between 0 and 1) counts from the corpus. For example in case of a bigram we can calculate the probabilities as below:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \quad \textbf{(1.18)}$$

Where C denotes the counts of the respective entities. For a complete N-gram based model the general counting based probability can be given as:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \quad \textbf{(1.19)}$$

1.4 **Smoothing**

One major problem with MLE is of Sparse Data. That is, in case of sparse N-Grams there are many N-Grams which do not occur in the corpus, hence producing 0 entries, which reduce the overall relative frequency to zero. Thus we use *smoothing* techniques to give these entries a small value instead of making them zero. There are many such techniques used, some of which are mentioned below:

1.4.1 **Laplace Smoothing**

This is also called the add-one smoothing as the constant 1 is added to all the N-gram counts in the corpus before these are converted into probabilities. This method generally does not perform very well as often too much probability mass is moved to the N-grams with zero counts. The adjusted counts,c_i^* for add-1 smoothing are defined as:

$$c_i^* = (c_i + 1)\frac{N}{N+V} \quad \textbf{(1.20) [1]}$$

Where N is the total number of word tokens and V is the vocabulary size (i.e. the total number of word types).

1.4.2 **Good-Turing Discounting**

Good-Turing Smoothing is based upon the idea that the probability mass to be assigned to N-grams with zero counts is estimated from the number of N-grams with higher counts. So the smoothed out count *c** of N-Grams with count *c*, will be estimated from the number of N-Grams with count *c*+1 (N_{c+1}) as below [1]:

$$c^* = \frac{(c+1)N_{c+1}}{N_c} \qquad \textbf{(1.21)}$$

where N_c is the number of N-Grams with count c. The weight assigned to the lower count N-Grams must then be discounted from higher count N-Grams using the same formula. Hence we calculate N_0, N_1; N_1 from N_2 and so on. In practice however, we only discount for *c* up to a certain *k* [4]. *k* is suggested to be 5 in [7]. So the corrected formula is [1]:

$$c^* = \frac{\frac{(c+1)N_{c+1}}{N_c} - \frac{c(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}} \qquad \textbf{(1.22)}$$

1.4.3 **Witten-Bell Discounting**

In Witten-Bell discounting the probability of unseen things (i.e. the N-grams with zero counts) is estimated from the probability of things seen at least once (i.e. the N-grams with non-zero counts). If *T* is the types that we have already seen and *V* is the vocabulary size (the number of types that we will ever see), then the adjusted counts are given as:

$$c_i^* = \begin{cases} \frac{T}{Z}\left(\frac{N}{N+T}\right), if\ c_i = 0 \\ c_i(\frac{N}{N+T}, if\ c_i > 0) \end{cases} \qquad \textbf{(1.23) [1]}$$

Where *Z* is the total number of N-grams with zero count.

1.4.4 Deleted Interpolation

Here the probability of the N-Grams with zero counts is estimated from the interpolated sum of the probabilities of the $(N\text{-}k)$-grams where (k = 1, 2, .., N-1). By using a weight λ the adjusted probability of a trigram can be given as:

$$P^*(w_n|w_{n-2}w_{n-1}) = \lambda_1 P(w_n|w_{n-2}w_{n-1}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n) \qquad \textbf{(1.24) [1]}$$

Such that: $\sum_i \lambda_i = 1$.

The values of λ can be trained using some EM algorithm or simply fixed to give more weight to higher N-grams as compared to lower ones. This however, depends on the importance that we wish to give to context versus simple occurrence of words (tokens) in the corpus.

1.5 The Training Phase

The α_{ij} and $b_j(o_t)$ matrices of the HMM are trained in the training phase of the ASR system. There are two main method of training available: *Hand Segmentation* and *Embedded Training*.

1.5.1 Hand Phone Segmentation

This is a relatively simple method (from the ASR system's point of view). The speech files are hand transcribed and also completely labeled regarding the starting and ending time of each words and phone in the utterances. Once such detailed information is available, the training of the α_{ij} and $b_j(o_t)$ matrices is just a matter of counting the occurrences of phones in the training data. The α_{ij} values are word specific while $b_j(o_t)$ are shared across multiple words which share common phones.

The problem with hand tagging of data is that it is an inaccurate and extremely lengthy procedure. It may take 400 hours to label 1 hour of speech recording [1]. The second reason i.e. loss of accuracy, means that humans are generally not good at doing phonetic transcription for units smaller than phones and also not very accurate at detecting phone boundaries. For these reasons it is often preferred to use the second method i.e. Embedded Training. The Hand Segmentation is often used for initial boot strapping of a system.

1.5.2 **Embedded Training**

In this method each phone HMM is trained while embedded in the entire sentence and the phone segmentation and alignment is done as part of the training. For this procedure the speech corpus is divided in small utterances. Then a transcription file, containing word transcriptions of these utterances in correct order, is constructed. A pronunciation lexicon establishes the mapping between words and phones and a phoneset contains all the possible (untrained) phones. The sentence HMM is built from this as shown in Figure 5, by using the Baum Welch as below:

i. A sentence HMM is built for each sentence, as shown in Figure 5.
ii. The α_{ij} probabilities are initialized to 0.5 (for loop back and next state transition) and all other transition probabilities are set to 0.
iii. The $b_j(o_t)$ probabilities are initialized by setting the mean and variance for each Gaussian to the global mean and variance for the entire training set. Steps ii and iii are termed as a *flat initialize*.
iv. Multiple iteration of Baum-Welch is run to train the system.

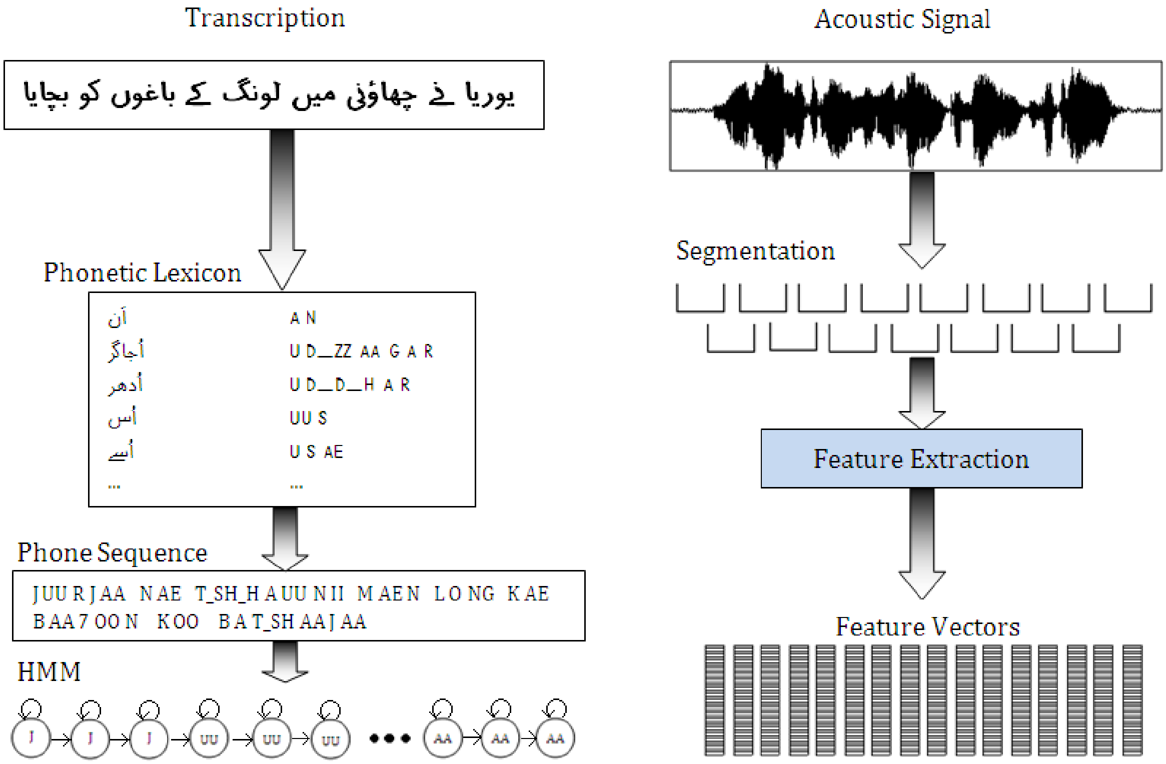


Figure 5 - Embedded Training [1]

The Baum-Welch repeatedly computes $\xi(t)$, the probability of being in state *i* at time *t*, by using *forward-backward* [1] to sum over all possible paths that were in state *i* emitting symbol o_t at time *t*. This allows the accumulation of counts for re-estimating the emission probability $b_j(o_t)$ from all the paths that pass through state *j* at time *t*.

1.6 The Decode Phase

Viterbi algorithm is used in the decoding phase [1]. In order to create a balance between the weights of likelihood and prior in Equation 1.6, we add a *language weight*, LW:

$$W' = \underset{W \in L}{\operatorname{argmax}} P(O|W)P(W)^{LW} \qquad \textbf{(1.25)}$$

The language weight is also referred to as the *Language Model Scaling Factor* (LMSF). It value is generally kept between 6 and 13, and an increase in LW causes a decrease in the value of LM probability (as it is between 0 and 1). So the goal of the Viterbi is to maximize equation 1.15.

The Viterbi is a dynamic programming algorithm. Given that it has already computed the probability of being in every state at time t-1, it computes the Viterbi probability by taking the most probable of the extensions of the paths that lead to the current cell. For a given state q_j, at time t, the value of $v_t(j)$ is computed as:

$$v_t(j) = \max_{1 \leq i \leq N} v_{t-1}(i) a_{ij} b_j(o_t) \qquad \textbf{(1.26)}$$

Where,

$v_{t-1}(i)$: The previous Viterbi path probability from the previous time step

a_{ij}: The transition probability from previous state q_i to current state q_j

$b_j o_j$: The state observation likelihood of the observation symbol o_t given the current state j

Where, the goal of the Viterbi is to find the best state sequence q = (q_1, q_2, ..., q_{T}) given the set of observations o = (o_1, o_2...o_T). It also needs to find the probability of this sequence, to do which the Viterbi takes MAX over the previous path probabilities. However, this tends to be slow for modern speech research. Therefore, for large vocabulary recognition we do not consider all possible words when the algorithm is extending paths from one column to the next. Instead the low priority paths are *pruned* at each step and are not extended to the next step. This pruning is usually implemented by using *beam search* in which at each time step t, the probability of the best (most probable) state/path D is computed. Then all the states worse than some threshold θ (*beam width*) from D are pruned. It is implemented with an active list of states that is kept for each step. Only transitions from these states are extended when moving to the next step.

1.7 **Evaluation**

The standard evaluation metric for ASR systems is the *Word Error Rate* (WER). The word error rate is calculated by comparing the output word string produced by the ASR system (called *hypothesis*) with the correct string expected from it (called the *reference*). The comparison is done by calculating the *Minimum Edit Distance* between the hypothesis and the reference. The minimum edit distance calculates the minimum substitutions, word insertions and word deletions required to map the hypothesis onto the reference. This minimum edit distance is then converted into WER as follows:

$$Word\ Error\ Rate = \frac{Insertions \times Deletions \times Substitutions}{Total\ words\ in\ the\ Reference\ String} \qquad \textbf{(1.27) [1]}$$

As the expression contains substitutions therefore the WER can exceed 100%.

Chapter 2

The Sphinx Speech Recognition System

This section briefly explains the architecture of the *Sphinx* Speech recognition system. The information has been extracted from the documentation available online and primarily from sources [8], [9], [10], [11], [12] and [13]. The Sphinx speech recognition system is an open source, high performance speech recognition system developed by the CMU *Sphinx project* that can be used in building speech recognition applications. It also includes related resources such as acoustic model trainer, language model trainer. Hence Sphinx is available as a complete speech recognition system trainer and decoder.

2.1 **Available Versions**

Several projects of sphinx are available. A brief description is given below:

- **Sphinx-2** is a high speed large vocabulary speech recognizer. It is usually used in dialogue systems and pronunciation learning systems. Sphinx-2 is the predecessor of *PocketSphinx*. It is not being actively developed at this time, but is still widely used in interactive applications. It uses Hidden Markov Models (HMM) with semi-continuous output probability density functions (PDF). Even though it is not as accurate as Sphinx-3 or Sphinx-4, it runs in real time, and therefore it is a good choice for live applications
- **Sphinx-3** is a slightly slower but more accurate Large Vocabulary Speech Recognition System. It is usually used as a server implementation of Sphinx for evaluation. It uses HMMs with continuous output PDFs. It supports several modes of operation. The more accurate mode, known as the "flat decoder", is descended from the original Sphinx-3 release. The faster mode, known as the "tree decoder", was developed separately. The two decoders were merged in Sphinx-3.5, though the flat decoder was not fully functional until Sphinx-3.7
- **Sphinx-4** is a completely rewritten version of Sphinx decoder in Java. It provides high accuracy and speed performance comparable to the state of the art. It uses HMMs with continuous output PDFs. Sphinx-4 uses models trained by Sphinx-3 trainer

- **PocketSphinx** is a speech recognizer which can be used in embedded devices. It is highly optimized for CPUs such as ARMs. PocketSphinx is CMU's fastest speech recognition system. It uses HMMs with semi-continuous output PDFs. Even though it is not as accurate as Sphinx-3 or Sphinx-4, it runs at real time, and therefore it is a good choice for live applications
- **SphinxTrain** is a suite of tools which carry out acoustic model training. SphinxTrain is CMU Sphinx's training package. It trains models in Sphinx-3 format, which is also used by PocketSphinx. The Sphinx-3 format can also be converted to Sphinx-2 format under some conditions related to Sphinx-2's limitations
- **CMU-Cambridge Language Modeling Toolkit** is a suite of tools which carry out language model training
- **SphinxBase** provides a common set of library used by several projects in CMU Sphinx.

According to the tests performed at CMU which compare Sphinx4 with Sphinx3 (flat decoder) and to Sphinx3.3 (fast decoder) in several different tasks, ranging from digits recognition to medium-large vocabulary, Sphinx3 (flat decoder) is often the most accurate, but Sphinx4 is faster and more accurate than Sphinx3 in some of these tests [8].

The decision about which version to use depends on the type of application being developed. A few considerations are:

Programming Languages

Sphinx-2 and Sphinx-3 are in C and Sphinx-4 is in Java. So these are all quite portable.

Accuracy and Speed

Very extensive benchmarking of different Sphinx versions is not available; however, the following rough estimates are present [8]. According to these measurements, the accuracy numbers can be summarized as:

Sphinx-4 ≥ Sphinx-3 > Sphinx-2 [12]

And in terms of processing time the comparison is as follows:

Sphinx-4 ≥ Sphinx-3 ≥ Sphinx-2 [12]

That means Sphinx-2 is still the fastest recognizer in the sphinx inventory. However its accuracy is also the lowest. Sphinx-3 is the best C-based recognizer which can be configured as both fast and accurate. Sphinx-4 is possibly the most all round recognizer and it can actually be very fast.

Interfaces

Sphinx-4 provides the best interface among all. It can be configured by using a configuration file and command-line. This advantage can make web development tasks much easier. Usage of Sphinx-2 and Sphinx-3 requires much more skill in scripting and in general understanding of the program. At the current stage, they are still regarded as systems for expert users.

Platforms

Sphinx-2, Sphinx-3 and Sphinx-4 are all platform-independent. However, the use of Java language in Sphinx-4 may allow higher degree of platform independence. It is also of common interest that whether any version of the decoders could be used in an embedded platform. Sphinx-2 is perhaps the best recognizer for embedded platforms due to its smaller size and lesser processing time.

Research

Sphinx-2, 3 and 4 all have a clean design and they all support continuous HMMs which is currently a de-facto standard of HMM. In the case of individual research, for acoustic modeling and fast GMM computation, Sphinx-3 is generally considered a better research platform for speech recognition. By itself, it supports Semicontinuous HMM (SCHMM), continuous HMM (CHMM) and Sub-vector quantized HMM. They represent three different kinds of modeling techniques for speech recognitions.

2.2 Sphinx-3

Sphinx-3 is the successor to the Sphinx-II speech recognition system from Carnegie Mellon University. It includes both an acoustic *trainer* and various *decoders*, *i.e.*, text recognition, phoneme recognition, N-best list generation, etc. The following is a brief summary of its main features and limitations:

- Works with *discrete*, *semi-continuous*, or *continuous* acoustic models

- Works with 3 or 5-state left-to-right HMM topologies
- Bigram or trigram language model
- Batch-mode or live operation from pre-recorded speech

This distribution has been prepared for UNIX platforms and can be ported to MS Windows as MS Visual C++ 6.0 workspace and project files have been provided.

2.2.1 **Sphinx-3 Trainer**

The Sphinx trainer trains the HMM Acoustic models for the recognizer. Figure 6 explains the working of the front end trainer system which converts audio files in Mel Frequency Cepstral Coefficients.

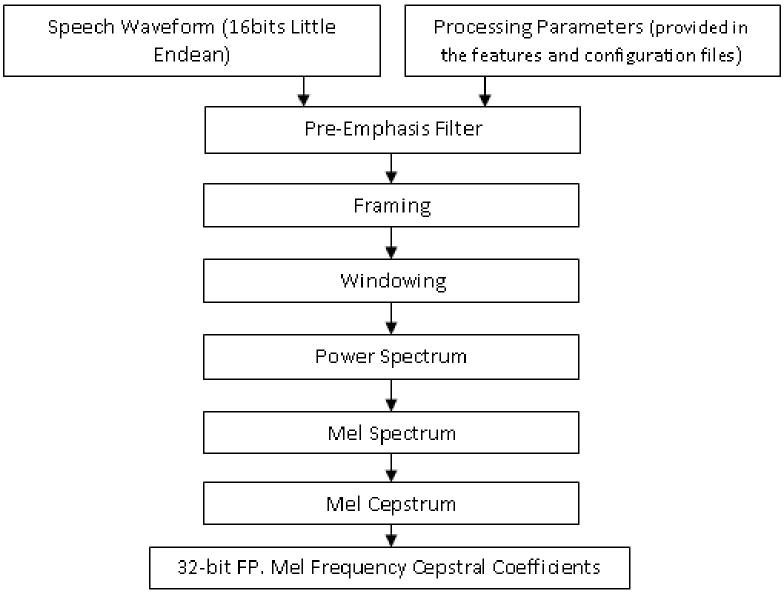

Speech Waveform (16bits Little Endean)
Processing Parameters (provided in the features and configuration files)
Pre-Emphasis Filter
Framing
Windowing
Power Spectrum
Mel Spectrum
Mel Cepstrum
32-bit FP. Mel Frequency Cepstral Coefficients

Figure 6 - Trainer Front-end [10]

Front End Parameters

A brief description of the front end parameters is given below:

- **Sampling rate:** The sampling frequency of the input speech signal
- **Frame rate**: The speed of how fast a window is moving. It is terms of number of samples

- **Window length**: The size of moving window in terms of number of samples. Hamming window is used
- **Number of Cepstral Coefficients**: The number of coefficient *including* the energy coefficient in the feature vector
- **Number of filters**: The number of filters that would be used in the filter banks
- **The size of FFT**: The number of points of FFT used
- **The *lower* filter frequency**: It is actually the lower frequency of the lowest filter in the filter bank
- **The *upper* filter frequency**: It is actually the upper frequency of the highest filter in the filter bank
- **The pre-emphasis coefficient**: The value of the pre-emphasis coefficient in the pre-emphasis filter

The MFCCs are then used to train the HMM Acoustic models by using embedded training approach and the Baum-Welch algorithm.

2.2.2 The Language Model Tool

The CMU statistical modeling toolkit can be used to generate the language models. It takes a text corpus as input and produces Trigram or Bigram Language Models in binary or *ARPA* format. These can be converted into binary dump file format by using the lm2dmp utility also (separately) made available with the Sphinx. The language model can be generated using any one of four smoothing strategies:

- Good Turing discounting
- Witten Bell discounting
- Absolute discounting
- Linear discounting

The SLM toolkit architecture is shown in Figure 7.

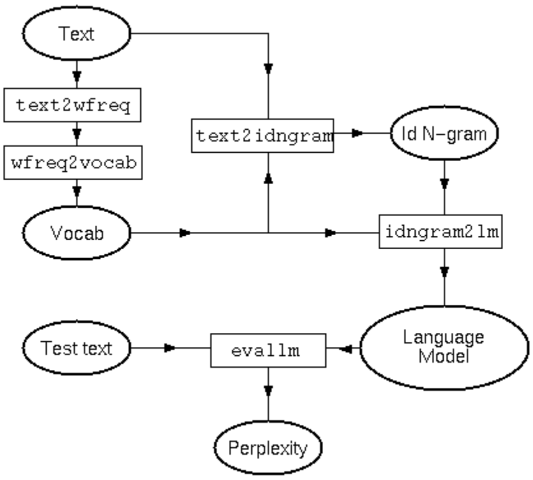


Figure 7 - SLM tool kit architecture [14]

2.2.3 **Sphinx-3 Decoder**

The Sphinx-3 decoder is based on the conventional *Viterbi search* algorithm and *beam search* heuristics. It uses a *lexical-tree* search structure. It takes its input from pre-recorded speech in raw PCM format and writes its recognition results to output files.

Inputs

The decoder requires the following inputs:

1. **Lexical model**

The lexical or pronunciation model contains pronunciations for all the words of interest to the decoder. Sphinx-3 uses *phonetic units* to build word pronunciations.

2. **Acoustic model**

Sphinx uses acoustic models based on statistical *hidden Markov models* (HMMs). The acoustic model is trained from acoustic training data using the Sphinx-3 trainer. The trainer is capable of building acoustic models with a wide range of structures, such as *discrete*, *semi-continuous*, or *continuous* HMMs.

3. Language model (LM)

Sphinx-3 uses a conventional back off bigram or trigram language model.

4. Speech input specification

Sphinx3_decode uses a control file for batch mode processing. The entire input to be processed must be available beforehand, *i.e.*, the audio samples must have been preprocessed into Cepstrum files.

Outputs

The decoder produces a *Recognition hypothesis*: A single best recognition result (or *hypothesis*) for each utterance processed. It is a linear word sequence, with additional attributes such as their time segmentation and scores. In addition, the decoder also produces a detailed log that can be useful in debugging, gathering statistics, etc.

Figure 8 depicts the detailed decoder operation. A complete description of all the files and file formats used by the decoder is given in section 5.4 and Appendix G.

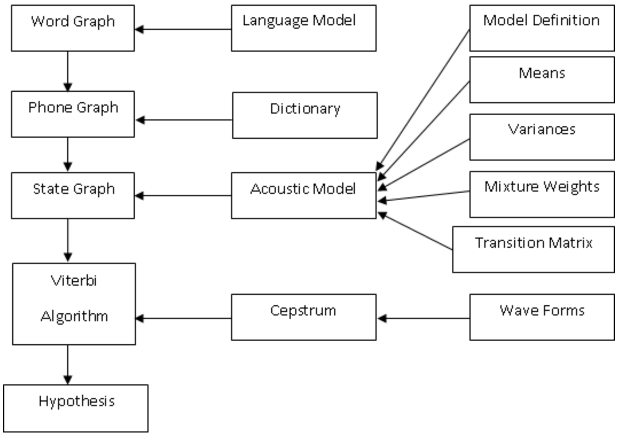


Figure 8 - Sphinx-3 Decoder [10]

Chapter 3

Urdu Acoustics and Letter to Sound Rules

Urdu, the national language of Pakistan, is spoken by more than a 100 million people around the globe [15]. Phonetically, it is a rich language with a large inventory of 44 consonants, 7 long oral vowels, 7 long nasal vowels, 3 short vowels and numerous diphthongs [16]. Let us briefly review the phonetic inventory of Urdu.

3.1 **Vowels**

Urdu has a rich variety of vocalic sounds. As shown in Figure 9 the Urdu vocalic sounds are distinguished on the basis of all the criteria of quality, duration and nasalization. Other than the the 7 long oral vowels, ɛ also occurs in Urdu, but only as a phone, not as a phoneme. Moreover, ɔ̃ also occurs as a phone and not as a phoneme. Except for this, all the other long vowels have nasal versions. The three short vowels in Urdu do not possess nasalized counterparts. The three higher back vowels also are characterized by lip rounding.

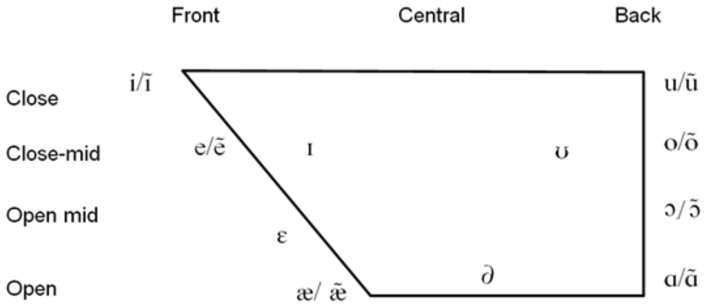
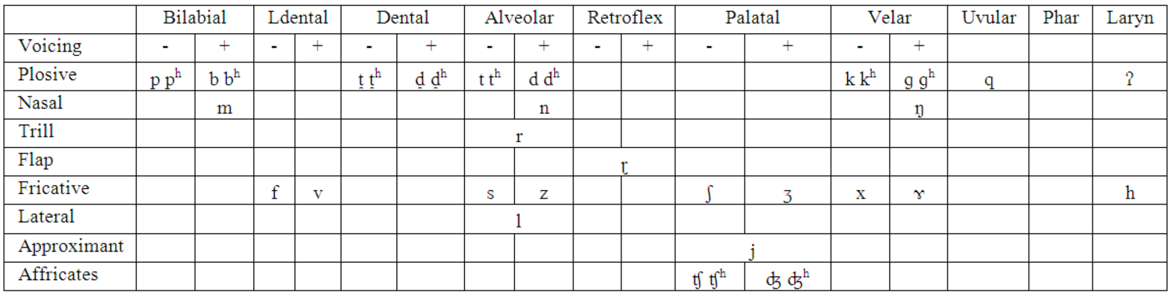


Figure 9 - Urdu Vowels [17]

3.2 **Consonants**

The 44 consonants of Urdu are shown in Figure 10. Once again we see a large variety of acoustic features with several examples of all the four voicing mechanisms of voiced, unvoiced, and voiceless and voiced aspirated sounds. Nasal consonants e.g. /m/, /n/ add to already diverse nasal sounds of Urdu, which mark a clear distinction between languages like English which possess relatively lesser number of nasal sounds.



	Bilabial		Ldental		Dental		Alveolar		Retroflex		Palatal		Velar		Uvular	Phar	Laryn
Voicing	-	+	-	+	-	+	-	+	-	+	-	+	-	+			
Plosive	p p^h	b b^h			t̪ t̪^{h}	d̪ d̪^{h}	t t^h	d d^h					k k^h	g g^h	q		ʔ
Nasal		m						n						ŋ			
Trill							r										
Flap									ɽ								
Fricative			f	v			s	z			ʃ	ʒ	x	ɣ			h
Lateral							l										
Approximant											j						
Affricates											ʧ ʧ^h	ʤ ʤ^h					

Figure 10 - Urdu Consonants [17]

Since we will be requiring phonetic transcription of Urdu words therefore, it is must that we briefly visit the letter to sound rules of Urdu. For a detailed discussion on the subject the reader may refer to [17].

3.3 Letter to Sound Rules for Urdu

It is written in Arabic script in Nastalique style using an extended Arabic character set [18]. The character set includes basic and secondary letters, aerab (or diacritical marks), punctuation marks and special symbols [19]. However, everyday-Urdu is written only using the letters, which primarily represent just the consonantal content, and the use of diacritics, which mostly represent the vowels in Urdu, is optional. Though this does not cause any difficulty for the native speaker, the absence of vowel marks makes the job of letter to sound mapping more difficult computationally [17]. As a result, Urdu corpora obtained from sources like newspapers etc. are generally phonetically transcribed using lexical lookup, though manual review is necessary for cases where multiple pronunciation are possible for same written form.

ْ	ّ	ً	ٰ	ُ	ِ	َ

ھ	ة	ں	آ

چ	ج	ث	ٹ	ت	پ	ب	ا
ز	ڑ	ر	ذ	ڈ	د	خ	ح
ع	ظ	ط	ض	ص	ش	س	ژ
ن	م	ل	گ	ک	ق	ف	غ
			ے	ی	ء	ہ	و

Figure 11 - Urdu Letters and Aerab [17]

The Urdu letters and aerab are shown in the Figure 11. These letters can be divided into the following categories on the basis of the different types of grapheme to phoneme mapping rules:

1. Consonantal characters

The characters shown in the Figure 12 always map to consonantal phonemes in Urdu. The mapping between grapheme to phoneme for these characters is many-to-one is some cases while one-to-one in others. However, it is not one-to-many in any case. Simple context independent mapping rules can be used to convert these graphemes to phonemes in any Urdu character string.

چ	ج	ث	ٹ	ت	پ	ب
tʃ	dʒ	s	ʈ	t̪	p	b
ڑ	ر	ذ	ڈ	د	خ	ح
ɽ	r	z	ɖ	d̪	x	h
ط	ض	ص	ش	س	ژ	ز
t̪	z	s	ʃ	s	ʒ	z
گ	ک	ق	ف	غ	ع	ظ
g	k	q	f	ɣ	ʔ	z
		ۃ	ہ	ن	م	ل
		t̪	h	n	m	l

Figure 12 - Letter to sound mappings of Urdu Consonants [17]

2. Characters which show dual (consonantal and vocalic) behavior

There are three characters in Urdu which depict the dual behavior of vocalic or a consonant depending on the context of occurrence. These characters are Alef (ا), Vao (و) and Yay (ے or ی). Vao changes to the voiced labiodentals /v/ and Yay changes to the approximant /j/ when these occur at the onset or coda of a syllable. While at nucleus positions they make long vowels.

3. Vowel modifiers

There is only one such example in Urdu and that is the letter Noon Ghunna (ں) which does not add any new sound but only nasalizes the previous vowel. If it occurs after a type 2 character, then it converts into nasal long vowel.

4. Consonantal modifiers

The Do-Chashmey Hay (ھ) acts as a consonantal modifier as it combines with stops, nasal stops approximants and affricates to form aspirated consonants.

5. Composite (consonantal and vocalic) characters

This category also has a single example, the Alef Madda (آ) which is just an alternative transcription for double Alef. It thus represents an Alef in consonantal and second in vocalic position.

Similarly the aerab can be divided into the following types:

6. Basic vowel specifier

There are short vowel aerabs used in Urdu called Zabar (◌َ), Zer (◌ِ) and Pesh (◌ُ). These combine with other characters to form vowels according to the following rules:

a. They make *short vowels* when they occur with type 1 and type 2 consonants and are not followed by type 2 letters

b. They generate *long vowels* when they occur with type 1 and type 2 consonants followed and combined by type 2 characters

c. They make *long nasalized vowels* when they combine with type 1 and type 2 consonants followed by type 2 characters followed by type 3 character i.e. Noon Ghunna

7. Extended vowel specifier

The single diacritic Khari Zabar (◌ٰ) is an extended vowel specifier, which represents an Alef. When it occurs on top of Vao or Yay, it converts the sound to Alef (/a/).

8. Consonantal gemination specifier

The Tashdeed or Shad (◌ّ) geminates consonantal characters except for Alef. As a result of the doubling the consonant acts as the coda of the previous syllable and the onset of the next one.

9. Dual (vocalic and consonantal) inserter

The Do-Zabar (◌ً) only occurs on Alef in vocalic position and converts the long vowel /a/ to the short Schwa followed by consonant /n/.

10. Vowel-aerab placeholder

This category includes Alef (ا), which is a letter and Hamza (ٔ) which is a diacritic. Alef occurs in this role at word initial positions while Hamza otherwise. At word initial positions Alef acts as a place holder for short vowel if no other consonant is there to act as one and the words starts with the short vowel. In words medial positions this role is taken up by Hamza in case of onset-less vowels.

Further if the preceding syllable ends on a vowel and next start with one then Hamza may be introduced between the two vowels. Lastly, if the preceding vowel is closed by a coda consonant, then Hamza may be used with Alef.

Chapter 4
Literature Review

A lot of work has been done on the development of Automatic Speech Recognition systems for many languages of the world. The work ranges from the activities involved in the development and enhancement of speech corpora to the development and improvements in the speech recognition systems. The main area on which I focused my study was the development of speech corpus, especially the ones for Asian languages, as I need to develop such a corpus for Urdu before being able to develop a speech recognition system. Secondly, as this work focuses on the development of an ASR system for continuous and spontaneous speech I have tried to include research work focusing on these areas in my study. Following is a brief review of the work done on Speech corpora development and Automatic Speech Recognition systems.

4.1 Corpus Construction

4.1.1 Speech Corpus for Amharic Large Vocabulary Continuous Speech Recognition System

Abate et al. [20] developed a Speech corpus for Amharic, the official language of Ethiopia, for training a large vocabulary continuous speech recognition system. Amharic speech contains 38 different phones with 31 consonantal and 7 vocalic sounds and at least 234 distinct CV (consonant-vowel) syllables. They used archives of a website where newspaper and magazine articles are published to build the corpus, which was cleaned semi-automatically. It consists of read speech only. The corpus has been phonetically enriched based on syllables. The syllable based phonetically rich corpus was collected using computational methods from a large corpus of around 100,000 sentences. It has been balanced by adding on the required phonetically rich content. The process of the phonetically rich corpus construction was divided into two steps. In the first step sentences with the highest phonetic richness were selected. Of these sentences, only the ones which preserve the syllabic phonetic balance above a certain threshold were kept. The syllable found to be missing in the final corpus, due to rare occurrence, were added by collecting the rare words required. These words were then converted into sentences according to the grammatical rules of Amharic. The corpus was divided into data for training, testing and speaker adaptation. The speech was recorded in office level background noise. The speech corpus contains 20 hours of training speech collected from 100 speakers who read a total of 10850 sentences. The sentences were split semi-automatically. The speakers represented all the five different dialects of Amharic. The total number of words present in the training speech amounted to 28666 that covered all 233 Amharic syllables. The corpus was annotated manually and split at word and syllable level.

4.1.2 **Speech Corpus for Turkish Text To Speech System**

Bozkurt et al. [21] designed a speech corpus for Turkish Text to Speech system based on read speech, and have used a Greedy algorithm for text selection. They have used diphones to simplify concatenation issues. The greedy algorithm used assigns costs to sentences according to the number of out-of-cover units (diphones) and in-cover units in the sentence. In each iteration the algorithm picks the sentence with maximum cost, i.e. which adds the maximum number of out-of-cover elements to the already made pool of sentences. It removes the sentence from the universal sentence pool and updates the target and universal sentence pools accordingly. They modified the algorithm for maximum variability of units, by using weighted sums instead of binary costs in the greedy decision making. They also used it to construct a Turkish speech corpus for TTS. The main tasks involved in the process of collecting the text for the corpus development included:

- Collection of Turkish text from the internet producing a total of 115000 sentences
- Preprocessing and rejection of sentences including some unexpected phoneme sequences primarily due to foreign words and mistyping
- Multiple iterations of the greedy algorithm and manual cleaning of the selected text which finally selected 2500 sentences
- Finally special sentences for including uncovered diphones were manually designed

4.1.3 Speech Corpus for Hindi Large Vocabulary Continuous Speech Recognition System

Chourasia et al. [22] have developed a speech corpus for Hindi to be used in the construction of a large vocabulary, continuous speech, multi-speaker recognition system which can also be used for recognizing fluent speech. They have chosen phonetically rich sentences from a Hindi corpus which was collected from various sources like hand typed articles, periodical, magazines etc. and text data available online. The data is automatically transcribed phonetically using grapheme to phoneme rules. The sentences chosen from the corpus must conform to certain requirements as mentioned below:

- The sentences should be short (with a minimum of 4 and maximum of 10 words)
- They are manually inspected to ascertain that they do not sound artificial
- They are meaningful
- They do not contain any offensive or sensitive words

The phonetic richness in the sentences is made context sensitive by making triphones as the unit of selection. Special attention has been paid to make sure that the rare phones and triphonemes are also sufficiently represented in the corpus. 350,000 sentences were chosen from the corpus and from these the phonetically rich sentences were chosen by using a program called "corpusCrt". 50,000 phonetically rich sentences were finally selected and divided into 5000 sets of 10 sentences each.

4.1.4 Cell Phone based American English Speech Corpus Construction

Heeman et al. [23] have presented the complete details of the task of the collection of cell phone based speech for the American English corpus SALA-II. The main goal of the corpus construction is to train speech recognition systems. The target community is the population of North, Central and South America and the focused speech transmission medium is the speech over cell phones. The speech had to be recorded from 4000 different speaker in different environmental conditions including cars, trains, buses, public places, busy streets, over speaker phones in cars and also while using car phone kits also in quiet offices. All nine accent regions and both the genders had to be covered. The conversations included 44 different read items including:

- City names, company names people's names
- Credit card numbers, dollar amounts, numbers, phone numbers
- Typical application words (e.g., stop, play)
- Time phrases, date phrases
- 9 different phonetically rich sentences
- 4 phonetically rich words. In order

Also included were several spontaneous item including:

- Speakers their first name
- The city they grew up in
- The current time and date
- Several yes/no questions

The phonetically rich sentences of English were collected from the Harvard corpus, the Timit corpus, and children's stories and constructed some sentences to enhance the phonetic richness. A total of 4412 such sentences were collected. After recording from 3200 speakers the phonetic richness of the recorded sentences was examined and the phones which lacked richness were included in the remaining 800 prompt sheets. SpeechView, a tool included in the CSLU toolkit, was used for the transcription of the speech data. They have reported speaker recruitment to be the most challenging part of the whole project.

4.1.5 **Phrase Based Phonetically Rich and Balanced Corpus for Mexican Spanish Language**

Villaseñor-Pineda et al. [24] present an automated method of building a phrase based phonetically rich and balanced corpus from the web, for Mexican Spanish Language. The hypothesis that they have tested is that the phonetic distribution of a corpus collected from the web will match that of the language. Or in other words the Web, due to its huge size, is already a phonetically rich and balanced source, and thus, taking a subset of it is enough to assemble a phonetically rich and balanced corpus. A smaller set of phrases is obtained on the basis of sentence perplexity measure in accordance with a language model. Lower perplexity sentences are kept as their phonetic distribution matches that of the language model. A comparison of the phonetic richness of the auto-selected web based corpus with that of the Spanish language showed that the hypothesis of the phonetic balance of the web based content was correct. The final corpus contained 864,166 words, and 20893 lexical forms spanning over a set of 6082 phrases.

4.1.6 Speech Corpus for Greek Dictation System

Digalakis et al. [25] have developed a dictation system for Greek. The recordings for the speech corpus were conducted on 55 male and 70 female speakers in three different environments: a sound proof room, a quite environment and an office environment. The speakers completed 291 sessions. Each session in the sound proof environment consisted of 180 utterances, each quiet session of 150 utterances and each office session of 150 utterances. 30 of the 180 utterances of the sound proof session were specially selected in order to contain rich phonetic coverage. Therefore the total number of utterances that were collected was 46,020. After cleaning the total collected speech amounted to 72 hours. 30 sessions were then recorded with spontaneous speech. For transcription the corpus was split in two sets. One set of 23,136 utterances was transcribed by the speech recognition group of the Technical University of Crete and the other set of 10,000 utterances transcribed by the Institute of Language and Speech Processing in Athens. The transcriptions were done for the speech and many types of non-speech events like mispronunciation, noise etc. The SRI's DECIPHER speech recognition system was used for the ASR. The gender independent model gave a Word Error Rate of 21.01%. the independent model for male speech gave 19.27% WER, while that for female speech gave 20.85% WER.

4.1.7 Speech Corpus for Automatic Transcription System for Spontaneous Dutch Speech

Binnenpoorte et al. [26] have developed a method for automatic transcription of Dutch spontaneous speech. A corpus of telephone conversations was used for the transcription tests.

All available transcribed data was gathered to act as the reference model. The continuous speech recognizer (CSR) was trained with nearly 24.5 hours of speech data containing 304502 words with around 14113 unique words. The test data consisted of 13 minutes of speech with 2850 words and 826 unique words. Using the CSR and a bigram language model the test data was transcribed. To test this automatically generated transcription the test data was manually transcribed by two phonetically trained and experienced listeners. They transcribed from scratch and were required to reach a consensus on each symbol in the transcription. They used the same symbol set as was used for the automatically generated transcription. This gave the reference transcription. An alignment of the auto-generated and reference transcription revealed 21.73% of phone error rate.

4.1.8 Microphone and Telephone based Russian Speech Corpus

TeCoRus speech corpus was collected by Ronzhin at al. [27] to with the goal of facilitating Russian speech research especially in the field of telecommunication. The speech corpus is designed to train context-dependent phone models. The speech has been recorded over two different channels: narrowband telephone channel (4 kHz, Moscow fixed telephone network) and broadband microphone channel (11 kHz computer Koss SB35 microphone). TeCoRus mainly consists of two parts. The phonetic part provides phonetically rich speech material to build the boot set of context-dependent phone models. This part contains 3050 utterances read by 6 speakers. The second part of the corpus is a collection of the short answers extracted from interviews and also includes both read and spoken material in the form of controlled answers and spontaneous speech on the predefined topic. It mostly consists of one-word positive and negative answers, digits (isolated and digit strings), numbers, names of months and weekdays, Russian first names, Spelling of Russian alphabet, time and date, few phonetically rich sentences and short stories. The second part was recorded from 100 speakers.

4.1.9 Minimal Corpus for Tamil, Telugu and Marathi Landline and Cellular Phone Based Continuous Speech Recognition

Anumanchipalli et al. [28] have used Sphinx-2 system for Tamil, Telugu and Marathi landline and cellular phone based continuous (read) speech recognition. Their system is speaker independent and is trained with the speech of around 560 speakers for the three languages. They started their work by developing a minimal text corpus that should cover all the phonetic variation that is present in these languages. The corpus consisted of phonetically rich sentences and the goal in its development was that it should also represent the syntactic structures of the target languages. The primary source for the corpora was newspapers and websites. Following (Table 1) is a summary of the corpora that they collected:

Language	No. of Sentences	No. of Unique Words	No. of words
Marathi	155541	184293	1557667
Tamil	303537	202212	3178851
Telugu	444292	419685	5521970

Table 1 - Summary of Corpora [28]

Grapheme-to-phoneme converters were used for the phonetic transcription of the corpora. Phonetically rich speech corpora for these languages were constructed using an “Optimal Text Selection (OTS) algorithm” that selects the sentences which are phonetically rich. They have utilized the greedy set cover algorithm for this task, which basis its selection on diphone based richness. The sentences are chosen to satisfy a certain threshold of diphone richness. The sentences were spoken by native speakers and 560 speakers were recruited to perform the task. The recordings were done on landline and cell phone channels. The models were trained and ASR systems for the target languages were developed. The following table (Table 2) lists the word error rates for the different systems trained:

ASR System	WER (%)	Vocabulary
Marathi Landline	20.7	21640
Marathi Mobile	23.6	18912
Tamil Landline	19.4	13883
Tamil Mobile	17.6	16187
Telugu Landline	15.1	25626
Telugu Mobile	18.3	16419

Table 2 - Result Statistics [28]

4.1.10 **Annotated Corpus for Spontaneous Chinese Language**

Li et al. [29] have developed an annotated corpus for spontaneous Chinese language and language effects. The Chinese Annotated Spontaneous Speech (CASS) corpus contains phonetically transcribed spontaneous speech. The goal of the corpus development was to capture the phonetic variations in Mandarin spontaneous speech due to pronunciation effects, including allophonic changes, phoneme reduction, phoneme deletion and insertion, as well as duration changes. The purpose is to develop an automatic speech recognition system for Mandarin casual speech. The recordings in the corpus are from sources like university lectures, student colloquia, and other public meetings. The environments for recordings included classrooms, amphitheatres, or school studios, resulting in a lot of background noise of different types. Initial around 6 hours of recordings were made. The speech was transcribed at words, syllable and semi-syllable level with detailed annotation of pronunciation variants. The detailed annotation of one hour of raw speech took around 380 hours of effort by a single transcriber. The speech was annotated in three tiers: the syllable tier, semi-syllable tier and miscellaneous tier. The CASS corpus is still under active development.

4.1.11 **Isolated Word Phonetically Rich Corpus for Lithuanian Speech Recognition System**

Raškinis et al. [30] have based the speech recognition system for Lithuanian on an isolated word phonetically rich corpus. HTK toolkit has been used for training the triphone based single Gaussian HMM speech recognition system based on Mel Frequency Cepstral Coefficients (MFCC). The best word error rate achieved was around 20% for speaker independent recognition of around 750 distinct isolated words. The training data included broadband recordings of 4 speakers (2 males and 2 females) of Lithuanian. Each recording consisted of 275 utterances that contained 2 to5 phonetically rich words. This gave a corpus containing 60.6 minutes of speech. Speaker adaptation and language modeling techniques were not used.

4.1.12 Comparison of Biphone and Triphone Rich Speech Corpora for Taiwanese

Yio et al. [31] have compared the performance of speech recognition engines for Taiwanese trained with two different speech corpora: one that covered all cross-syllable biphones and another which covers all cross-syllable triphones. The triphone rich corpus requires 10 times more words then the biphone rich one. A greedy algorithm is used for the word set construction. In order to collect the corpus data with as much phonetic richness as possible while keeping the words small a two step process was followed. First two sets of words were selected such that they covered all the cross-syllable bi-phones and tri-phones. In cross-syllable biphone words set, total distinct cross-syllable bi-phones were selected before total distinct syllables. In tri-phone words set, total distinct cross-syllable tri-phones were selected before the total distinct syllables. All the words were recorded by a male Taiwanese speaker over a microphone. The resulting speech corpora consisted of 100 minutes duration for each of the 10 sets of biphone rich words. The corpora were used to train speech recognition systems. Three different types of training data were used with durations ranging from 100 to 167 minutes. The training data was composed of 18.4 minutes of speech recordings. It was found that the Syllable Recognition Rate for Inside Syllable Biphone rich corpus reached 95.38% and that of Inside Syllable Triphone rich corpus reached 93.08%. Similarly the Syllable Recognition Rate for across syllable bi-phone corpus reached 95.75% while that of cross syllable triphone rich corpus reached 94.48%. In both cases the recognition rates for triphone rich corpora are less than the biphone rich corpora. The authors have attributed this discrepancy towards scarcity of training data. So their results show that the triphone rich corpus does not show significant advantages over the biphone rich corpus.

4.2 Corpora Construction Summary

A lot of work has been done on the development of speech resources for many languages of the world. These resources have been developed both for TTS (e.g. [21]) and ASR systems (e.g. [20], [28], [22] and [25]). The main goal in the development of speech corpora is phonetic coverage [27], which allows them to represent the phonetic structure of the target language. Speech corpora have been developed for various tasks, including: (a) isolated word corpora, e.g. Lithuanian [30], (b) continuous speech, e.g. Indian Languages [28], Hindi [22] and Greek [25], and (c) continuous and spontaneous speech, e.g. Dutch [26], Mandarin [29], and Russian [27].

The second criterion for the speech corpus development is the phonetic balance, i.e. the phonetic content should occur in the same proportions as in the language, to properly train the statistical models, as discussed for Russian [27], Amharic [20], and Mexican Spanish [24]. The phonetic richness can simply be phone-based [29] or context-based. The context-based methods take into consideration either a single immediate context, using diphone-based methods [28] or both beginning and ending context, using triphone-based methods ([22], [27]). However, an analysis in [31] shows that the triphone richness may not improve the accuracy of speech recognizer significantly but it requires much more data.

There is also difference in approaches towards gathering the data for the speech corpora. Most of the automatic approaches utilize some kind of a greedy algorithm to maximize the number of sound units (half-phones, phones, diphones or triphones) in minimal data set ([28], [21] and [31]). Still other make phonetically balanced sentences by comparing the phonetic composition with a language model by using perplexity [24]. This set is made richer by adding spontaneous speech data, e.g. from interviews [27] or recorded free speech. Still other approaches may include collection of text which represents the phonetic richness and proportion of a language [24].

4.3 Design of ASRs for Spontaneous and Continuous Speech

4.3.1 Speech Recognition System for Spontaneous Germen Speech

Sloboda et al. [32] developed a speech recognition system for spontaneous Germen speech, using Janus 2 speech recognition system, by developing an acoustic database driven approach. In case of spontaneous speech phenomena like false starts, human and nonhuman noises, new words, and alternative pronunciations pose a challenge in recognition. Therefore, the actual required pronunciation (phonemic pronunciation) of a word cannot be considered as recognition criteria. Rather the pronunciation should be chosen by the frequency with which it appears in the speech database. Therefore, in their approach they add new pronunciations to the database on the basis of the gathered speech data, especially for frequently misrecognized words. They have argued that the popular methods of using the intended pronunciation as the recognition criteria is not useful in case of spontaneous speech as words are seldom pronounced to perfection. However, modifying the phonetic dictionary by hand is also not a very feasible or useful approach as when the number of phones increases it is difficult to maintain consistency in manual transcription. Also the actual pronunciations may not be easy to predict without analyzing the actual data, e.g. in case of the pronunciation of foreign words and names. In addition it may not be easy to determine which acoustic variants are statistically more relevant. Therefore, to solve this problem they propose a computational technique in the form of an algorithm to improve phonetic dictionaries. This algorithm should optimize dictionaries on the basis of recognition performance and statistical relevance of pronunciations. The training and test data details are shown below (Table 3):

	Training Data	**Test Data**
No. of Dialogues	608	8
No. of Utterances	10735	110
No. of Words	281160	2346
Vocabulary size	5442	543

Table 3 - Training and Test data summary [32]

They achieved best word accuracies (WA) between 65.6% and 68.4% for the multi speaker German Spontaneous Speech.

4.3.2 Automatic Detection of Pauses in Spontaneous Japanese Speech

Masataka et al. [33] have developed a method for automatically detecting pauses in spontaneous speech which result in errors in the speech recognition. Their work is focused on Spontaneous Japanese speech. Catering for the pauses that occur in natural spontaneous speech, allows for recognition of more naturally spoken continuous and spontaneous speech. Their approach assumes that speakers do not change articulatory parameters during filled pauses. In their paper they have concentrated on two phenomena of spontaneous speech, word lengthening and filled pauses. Filled pauses are vocalized pauses which occur when a speaker hesitates or plans for further speech. They have proposed a detection mechanism for filled pauses on the basis of their spectral properties. As mentioned before they assume that the speaker does not vary the vocalic acoustic parameters during the vocalized pauses. Hence these pauses are characterized by a slight change in F0 and a small deformation of spectral envelop. The amount of F0 change and deformation of spectral envelop is hence used as an indicator for the possibility of filled pauses which is then evaluated probabilistically. They have presented their results of the correct detection of filled pauses tested on spontaneous speech corpus consisting of 100 utterances by five men and five women, each containing at least one filled pause. The recall rate i.e. *the number of filled pauses detected correctly/the total number of filled pauses* they have achieved is 84.9%. They applied their methods to the word alignment in HMM based speech recognition however they have not reported the improvement in Word Error Rate achieved.

4.3.3 Performance Comparison of Spontaneous Speech Recognition Systems vs Dictation Systems

Jacques et al. [34] explain the poor performance of spontaneous speech recognition systems compared to dictation system on the basis of inadequate language models. This is because of a lack of training data modeling the spontaneous speech disfluencies. They propose an approach for improving spontaneous language models by flexibly manipulating the context when disfluencies occur. They express that the WERs for large vocabulary speaker independent dictation systems is around 5%, while that of spontaneous speech recognition goes to 15% for broad cast news ([35] and [36]) and 40% for meeting and telephone conversation transcription [37]. They have categorized disfluencies into hesitations, repetitions and sentence restarts. In their method the context in an N-gram language model is modified as disfluency occurs. Thus they proceed to generate models for all the three types of disfluencies. They tested their system for telephone based spontaneous American English Speech using the Switchboard Corpus and ESAT Speech Recognition System. The acoustic models were trained on 310 hours of speech data. The trigram language model was generated from 3 million words of the Switchboard transcription corpus smoothed using Good-Turing Discounting. The lexicon consisted of 27000 words. The test data consisted of 20 telephone calls (almost 2 hours of data), containing 1718 sentences with 20K words. The tests showed that the method reduced the WER from 29.8% (without using the technique in this paper) to 29.6% after applying the proposed improvement.

Since only about 20% of the sentences have disfluencies so to get a better idea the experiment was repeated with only the sentences with which the proposed method changes the recognition results. The WER reduced from 36.7% to 35.1% in case of repetition however became worse for other types of disfluencies.

4.3.4 Technique for Paraphrasing Spontaneous Japanese Speech into Written Style Sentences

Takaaki et al. [38] proposed a method for paraphrasing spontaneous Japanese speech into written style sentences. As there is a lot of difference between spontaneous and written style Japanese most other speech recognition systems do not perform very well. Therefore they have proposed a method to translate spontaneous speech directly into written text by using a Weighted Finite State Transducer (WFST). The spontaneous speech transcribed by normal speech recognition systems is difficult to read due to disfluencies, filled pauses, repetitions, repairs and word fragments. Moreover the spoken Japanese has a much different style then written one e.g. Because of change of verbs, auxiliary verbs and adjectives etc. The proposed method is useful in tasks like automatic generation of captions, minutes, annotations etc. in their method they compose two WFSTs one for recognition and another for paraphrasing. The combined models are better than separate ones as this allows simultaneous translation of speech into readable text and also the linguistic constraints in speech recognition are reinforced as the two models work together. The method is tested using a 20,000 words spontaneous Japanese speech in the form of lectures. The Language Model was derived from a 36.8 million word corpus from newspapers and World Wide Web. The experiments showed a decrease in error rate ranging between 2.2 and 5.3 percent. The minimum WER for recognition achieved was 24.2% and maximum was 39.1% (for different test data). The decrease in WER in paraphrasing was between 2.2 and 5.2 percent, as manually transcribed data was compared with the automatically paraphrased data. The WER ranged between 29.6% and 51.5%.

4.3.5 **Disfluent Repetitions in Spontaneous Speech**

Vivek et al. [39] have concentrated on the problem of disfluent repetitions in spontaneous speech. They have used a metric called Repetition Word Error Rate (RWER) to quantify the errors caused by this type of disfluencies. They have proposed a solution by using an acoustic prosodic classifier for these disfluencies and a multi word model for modeling repetitions. Using this approach they have analyzed the RWER in the Fisher's conversational speech corpus. The classifier approach did not work very well as it produced many unnecessary warnings, however the multi word model approach resulted in the absolute RWER reduction of 1.26% and an absolute WER reduction of 2% on already well trained acoustic and language models. The data used for testing is of 5849 conversations from the Fisher's corpus, each lasting up to 10 minutes. They used 20 hours of data for training the system from the Fisher's corpus. The test corpus consisted of 2 hours of data from 20 speakers not overlapping with the training data. The percentage of repetitions in the test set was 1.91%.The language models were constructed from the transcribed training data transcripts and the remaining part of the Fisher corpus and also other conversational data sources. The classifier was generated from the repetitions in the training data by using features like duration, F0 and pause information from the boundary of the repetitions. The pause information after the repetition and F0 values around boundary were taken into consideration as were the duration information and the creakiness of voice in the repetition. However, the classifier generated lot of "false alarms". The word error rates before the multi word training remained within ranges of 42.1% to 48% with different acoustic models and between 48% and 56% for different language models. Next the system was trained with multiword models in which the frequently repeated words were trained as multiwords and also added to the dictionary. An improvement of 2% in WER resulted by incorporating these models into the training system.

4.3.6 **An Unsupervised Approach towards the Transcription of Continuous Broadcast News Data for Training Continuous Speech Large Vocabulary Speech Recognition Systems**

Frank et al. [40] have proposed an unsupervised approach towards the transcription of continuous broadcast news data for training continuous speech large vocabulary speech recognition systems. Since a large amount of transcribed acoustic data is must for training a good speech recognition system, therefore speech from various sources is generally transcribed manually, which is a very lengthy and painstaking process. This paper proposes that such raw (untranscribed) acoustic data can be transcribed by using an already trained speech recognition system, referred to as unsupervised learning because the system is not given the information about the correct output. The authors have tried two methods. One in which a speech recognition system trained with one to six hours of speech data is used to transcribe seventy two hours of speech. The complete data is then used to develop a speech recognition system. Next the system is trained iteratively starting from only one hour of transcribed speech data. In this approach the initial (small) speech recognition system is used to transcribe the whole seventy two hours of speech. Next the speech recognition system is trained with this speech and again used to transcribe the seventy two hours of speech, giving better results as the model had been improved. A confidence score is use to restrict the training to those portions of the speech corpus where the words are most probably correct. This was repeated several times. With the iterative approach the Word Error Rate was reduced from 71.3% to 38.3% on the Broadcast New' 96 evaluation test set and from 65.6% to 29.3% on the Broadcast news' 98 evaluation test set. When compared to a manually transcribed speech system for seventy two hours of speech the word error rate increased by 14.3% for the Broadcast New' 96 evaluation test and 18.6% the Broadcast news' 98 evaluation test using gender independent within-word models. However, using across-words models the word error rates increased by 20.9% and 23% respectively.

4.3.7 Automatic Transcription System for Arabic

Soltau et al. [41] have reported the outcome of a project to transcribe Arabic broadcast news using the GALE ASR. The training data used for the procedure consisted of 85 hours of broadcast speech data with transcripts, 51 hours of speech data from another source with transcriptions and 1800 hours of unsupervised data without transcriptions. Other resources used for training included the Arabic Gigaword corpus and transcripts for Arabic from various sources including a 28 million word resource. A 4-gram language model with 58 million n-grams was trained with Kneser-Ney smoothing. The system was tested on nearly 13 hours of speech from broadcast news and TV shows. The main problem faced was due to the lack of diacritics in everyday Arabic. The solution to this problem utilized was to train the initial models using fully hand diacritized transcribed speech. Then using this initial training model performs unsupervised learning of the non transcribed data. That ensured diacritization of the overall data. The Word Error Rates obtained ranged between 14.9% and 25.1% for different types and sizes of test data. This means an overall 42% relative decrease in Word Error Rate.

4.3.8 **Automatic Speech Recognition System for Spontaneous English Speech using the English CMU Recognition Dictionary**

Nedel et al. [42] have developed an ASR for spontaneous English speech using the English CMU recognition dictionary and the Sphinx-3 speech recognition system. They argue that in case of spontaneous speech, contrary to carefully articulated speech or read speech, the uttered sounds hardly ever conform to linguistic assumptions and rules regarding pronunciation. Therefore, there may be patterns of acoustic variations among single phones. In that case it is beneficial to model these variations of phones as separate phones. They have tested their hypothesis by applying such a phone splitting model to two phones AA and IY. They proposed three different methods for phone splitting and applied it to all the instances of these phones in the CMU dictionary. The speech corpus used is the NIST Multiple Register Speech Corpus for spontaneous speech with parallel transcription. The training data consisted of 2 hours of spontaneous speech and test data composed of 0.5 hours of continuous speech. The baseline performance of the system without applying the phone splitting came out to be 51.1% in terms of Word Error Rate. By applying the Gaussian Splitting technique on the said phones the WER reduced to 49.6% for splitting AA, 49.3% for splitting IY and 50.2% for modeling AA and IY as split phones. In case of their second technique the HMM likelihood based phone splitting the performance did not improve over the baseline for splitting AA, became worse after splitting both however, gave an improved WER of 49.6% after splitting IY. In the third method, duration based phone splitting was used. In this case the performance improved in all three cases. In case of splitting AA it went to 49.8% WER, in case of IY, 49.6% WER and finally after modeling both AA and IY as split phones the WER reduced to 49.9%.

4.3.9 Speaker Independent Continuous Speech Recognition System for Transcribing Unrestricted American English Broadcast News

Gauvain et al. [36] developed a speaker independent continuous speech recognition system for transcribing unrestricted American English broadcast news, with a best WER of 20%. The acoustic models were trained with around 150 hours of audio data. The language models were obtained by interpolation of backoff n-gram language models trained on different text sets including 203 million words of Broadcast News transcriptions, 343 million words of newspaper texts and 1.6 million words corresponding to the transcriptions of the Broadcast News acoustic training data. The phonetic lexicon contained 65,122 words and 72,788 phone transcriptions. A system of 48 phones was used with additional units representing silence, filler words and breath etc. On a test data of around 3 hours a word error rate of around 20% was achieved.

Chapter 5
METHODOLOGY

5.1 Introduction

The main difference between the work done towards the development of an Automatic Speech Recognition System in this thesis and similar works being done for other languages is the lack of phonetically transcribed corpora for Urdu. Therefore a goal of this effort was to develop such resources for Urdu and then to utilize them to develop a working speech recognition system. So the objectives of the thesis can be summarized as below:

- To develop a speech corpus for continuous Urdu speech (read speech corpus)
- To develop a speech corpus for continuous and spontaneous Urdu speech (spontaneous speech corpus)
- To utilize the speech corpora to develop a speaker specific Automatic Speech Recognition system

In order to achieve these goals the work was divided into separate parts. Each targeted to achieve a partial goal. The final product is the combination of all the subparts. These steps can be summarized as below:

1. **Design of a Speech Corpus**
   a. For continuous speech, a text corpus has to be developed that can be read and recorded
   b. For spontaneous speech interviews have to be designed to capture spontaneous dialogue based speech
   c. Various tools and techniques have to be developed to facilitate the development and evaluation of these resources
2. **Development of the Speech Corpus**
   a. The text corpus has to be read out and recorded in controlled environmental condition

b. The spontaneous corpus has to be generated from interviews and question-and-answer sessions

c. Both these corpora are analyzed into separate utterances

d. The utterances of the spontaneous corpus have to be transcribed in Urdu and also using phonemic transcription

3. **Adaptation of the Speech Corpus to the Speech Recognition system**

   a. This step requires processing the speech corpora to conform to the input standards of the Sphinx speech recognition system

   b. This step requires the development of a compiler like tool which takes a transcribed corpus as input and produces all the resources required by the Sphinx ASR system as output

4. **Training and testing of the speech recognition system**

   a. The main task involves the study of the internal working of the ASR and to fine tune all parameters to produce satisfactory results

Following sections discuss the methodology adopted in achieving all these goals.

5.2 Design of a Speech Corpus

The Large Vocabulary Automatic Speech Recognition (LVASR) system for Urdu requires the construction of a *phonetically rich* and *balanced* speech corpus for recognition of continuous and spontaneous speech in Urdu. Here phonetically rich means that the corpus should cover all the phonemes (and hence phones) present in Urdu. And *balanced* refers to the property that the phonemes should occur in the corpus with the same relative frequency distribution as in naturally spoken Urdu. The first part of this task is to develop a text corpus with these properties. That corpus can then be read out by a native speaker of Urdu and recorded to produce the Speech corpus for continuous Urdu speech.

So the first goal is to develop a sentence based corpus for Urdu, automating the design task as much as possible using existing language resources. The fundamental criterion remains to cover all possible phone combinations that are used in Urdu. The resulting phonetically rich corpus can serve to provide the baseline acoustic models for continuous LVASR for Urdu. However it will not necessarily be phonetically balanced. In order to convert it into a balanced corpus, recordings of actual interviews, using everyday speech will be done and transcribed. This will not only serve to balance the corpus but also model the spontaneous speech.

The first step is to find all possible phones that are used in Urdu Speech. A more practical approximation is to find all possible phonemes that exist in Urdu and then try to construct a word based corpus with the goal of covering all those phonemes. The word list is then manually converted into sentences. The Urdu corpus that is used for this purpose has been developed at the *Center for Research in Urdu Language Processing* (CRULP, [43]) and consists of 18 million words of Urdu. This data is gathered from various domains. The corpus is not fully diacritized and hence cannot be mapped completely to phonemes using simple letter to sound rules [17].

It must be mentioned here that an approach could have been to pick phonetically rich sentences directly from the corpus instead of making a word list and then converting them into sentences. However, this sentence list would not have been *minimal*, a criterion that can be controlled while making a word list. It would have been more natural representation of the language, even if redundant, but it has not been possible (especially for vowels) because the available Urdu text corpus is not clean and diacritized.

5.2.1 **Corpus Analysis and Development of Lexicons**

The available resource included an 18.2 million words corpus. This resource had been gathered from sources including newspapers and online pages related to sports, politics and current affairs. This corpus however was not clean, diacritized and/or normalized and the cleaning process required much manual intervention. However a word to frequency list for all the corpus words and a phonetic lexicon in SAMPA was also available for this corpus. The phonetic lexicon consisted of around 54,000 unique words. The second resource available was the *URDU (Pakistan) 200 SPEAKER DATABASE* developed by the *Appen Pty Ltd* [44]. This consisted of 3373 sentences comprising of 4336 unique words. The two corpora were merged and this resulted in an overall lexicon with 56000 unique words of Urdu. This was done to ensure that most of the common Urdu words are present in the lexicon. A list of these words was formed and the new words were phonemically transcribed in two passes. In the first pass an automatic transcription was done using the letter to sound rules. However, due to the lack of diacritics this resulted in partial transcription only. In the second pass the words were manually phonemically transcribed. As a result a complete phonetic lexicon is obtained which gives word to phoneme set mappings (henceforth referred to as *phonetic lexicon*). SAMPA representation [45] is used for phonetic and phonemic transcriptions. However, it was later converted to a format that will be referred to as *Case Insensitive SAMPA* (CISAMPA) in this text. This was done to match the requirements of the Speech Recognition Engine used in this research. The details of the new CISAMPA format and its motivation are given in Section 5.4.1 and Appendix A.

Next a word frequency analysis of the corpus is done to find the frequency of occurrence of all the 56,000 words in the corpus. This analysis gives another lexicon containing word to frequency mappings (henceforth referred to a *word-frequency lexicon*).

Summary of Language Resources available:

- Urdu text corpus (this will be referred to as the *Main Text Corpus*) with 18.2 million words. This not clean and diacritized, however, word to frequency lists and phonetic lexicon for it were made available to me
- Phonetic Lexicon for the Main Text Corpus
- Words to frequency list for the Main Text Corpus
- The 200 Speaker Database Corpus with 4336 unique words, of which around 2000 were not already present in the phonetic lexicon of the Main Text Corpus

Summary of Language Resources developed:

- A combined Phonetic Lexicon with around 56000 entries all of which are completely phonetically transcribed in CISAMPA
- A combined Word-frequency listing with around 56000

5.2.1.1 Tools Developed

In order to accomplish these tasks Java based text processing software was developed, which allowed the following functionality:

- **Inputs**:
    - Combined Urdu text corpus in Unicode (16-bit Big Endean) format (The 200 Speaker text corpus was analyzed using this)
    - Phonetic lexicon
    - Letter to Sound rules of Urdu transcription
    - SAMPA to CISAMPA mapping rules
- **Functions:**
    - *Corpus Cleaner:* Removes white space, punctuation marks etc. from the input corpus. Also normalizes the input text to a certain extent. Generates a list of words not already present in the Phonetic lexicon. This can be effectively used with the 200 Speaker corpus as it is relatively small
    - *Letter to Sound Converter:* Converts the input wordlist to SAMPA using letter to sound rules (assuming the input ios fully diacritized). This facilitates the transcription process. The 2000 words of the *URDU (Pakistan) 200 SPEAKER DATABASE* were transcribed with the help of this tool
    - *SAMPA to CISAMPA:* Converts input files in SAMPA format to CISAMPA formats
    - *Exception List Generator:* Generates a list of all the words in the corpus that are not present in the lexicon and transcribes them using letter to sound rules and allows manual modification, hence, it helps in the lexicon development

5.2.2 Phonetically rich word list

The primary goal of the development of a phonetically rich corpus was to ensure that it represents all sounds that occur in Urdu. This allows it to serve as a baseline for training an Automatic Speech Recognition system. However since the targeted ASR system is to be used for continuous and spontaneous speech, therefore a simple occurrence of all phonemes will not suffice for the following reasons.

5.2.2.1 Effects of phonetic context

The acoustic properties of a phone are not localized and are affected by the acoustic properties of the neighboring phones, i.e. the phonetic context. An example of this phenomenon is depicted in Figure 13 where the spectrogram plots of phone [**b**] are shown as it occurs between vowels [ɑ] and [**i**]. The effects of the context ([ɑ] and [**i**]) are quite pronounced on the target phone [**b**] through the formant transitions into and from the closure. As a result of this the different phones must be present in the Speech corpus while occurring in different phonetic contexts.

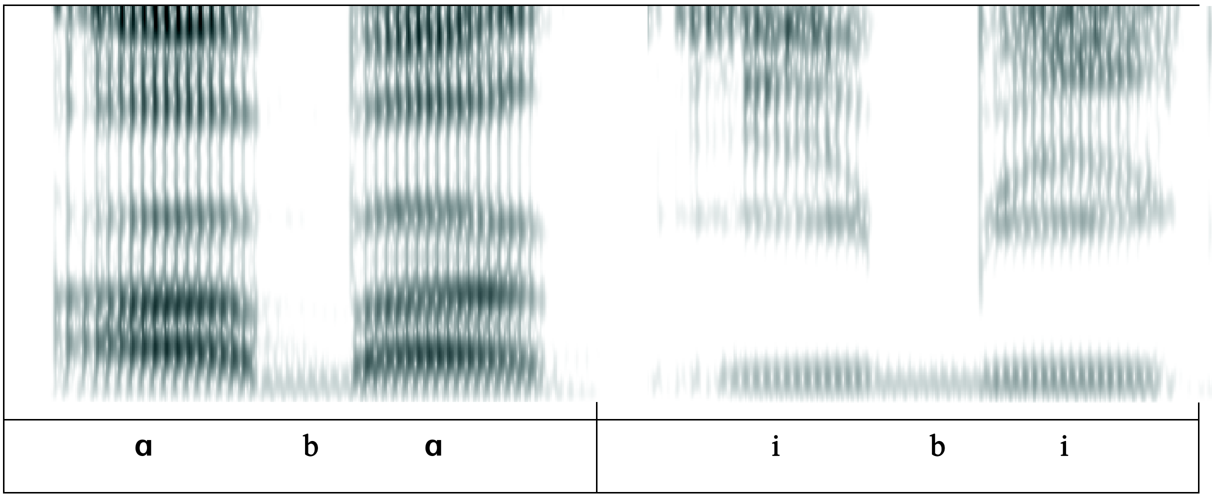


Figure 13 - Effects of phonetic context

5.2.2.2 Across Word Effects

In continuous speech, words run into each other hence the last phone of a word maybe affected by the initial phone(s) of the following word. Hence across-word influences will form a part of the phonetic context as well.

5.2.2.3 Spontaneous Speech

As mentioned earlier, the system must be trained for spontaneous speech in which the words are not carefully articulated. This often results in shorted words with missing phones or modified versions of target phones. Hence, it is required that the system should be trained by a model coming from free speech as well. However, this should be in addition to the speech read from phonetically rich text, as spontaneous speech is not guaranteed to be phonetically rich.

5.2.3 **Tri-phoneme based phonetically rich corpus**

As mentioned in 5.2.2.1, simple phonemic enrichment cannot guarantee that the resulting word set will be a representative of the acoustic properties of all the phones. Hence, context must be added to the phone set. For this we made sequence of three phonemes (henceforth referred to a *tri-phoneme*) the basic unit of the acoustic model for a particular sound. We define a tri-phoneme to consist of three phonemes {$\boldsymbol{P_1}$ $\boldsymbol{P_t}$ $\boldsymbol{P_3}$}, where $\boldsymbol{P_t}$ is the target phoneme and $\boldsymbol{P_1}$ and $\boldsymbol{P_3}$ act as the phonetic context. Now in order to represent the acoustic properties of all sounds, the dataset should contain all possible tri-phonemes that can occur in the language.

As the phonemic inventory of Urdu language comprises of 62 phonemes (excluding silence), there can be a total of 250,047 potential tri-phoneme combinations (including silence as a phoneme). In order to find the tri-phoneme combinations that actually exist in Urdu the phonetic lexicon is analyzed for tri-phonemes and their frequency of occurrence. This analysis is done from two different perspectives.

The first analysis is done to find all the unique tri-phonemes that occur within words. It is assumed for this analysis that all words are followed and preceded by silence, which is dealt with as a separate phoneme represented by the symbol #. The analysis shows that the corpus contained 18,294 unique in-word tri-phonemes.

Next, an analysis of the across-word tri-phonemes is performed. However, for this analysis, instead of finding all the *existing* across-word tri-phonemes, all the *potential* across word tri-phonemes are found (including the in-word tri-phonemes). This is done due to the flexible syntactic structure of Urdu which allows many possible arrangements of words in a sentence. This is achieved by assuming that every word (preceded by silence #) can be followed by any other word (followed by silence #). This list is found to contain around 85,000 unique tri-phonemes. This number should be interpreted as the upper limit of the number of tri-phonemes in Urdu (if vocabulary is limited to the used lexicon).

5.2.3.1 Word list construction

In order to allow utility in an ASR system the word list used for training the speech model must consist of a minimal word set that can maximize the number of tri-phonemes. In order to compute the word set a modified version of the Set Covering algorithm [46] is used. A decision has to be made whether to maximize the number of across-word tri-phonemes or the in-word ones. If the list is constructed on the basis of across-word tri-phonemes the word set will consist entirely of word-pairs instead of words. This will make the job of converting these into sentences very difficult (and for some combinations of words it will not even be possible).

Moreover, even if the sentences are generated from a wordlist based only upon in-word tri-phonemes, they will necessarily contain many of the across-word tri-phonemes and a post analysis can reveal the shortcomings. These can in turn be compensated by generating a supplementary sentence list. Hence, the wordlist is generated using the in-word triphonemes only.

5.2.3.2 Minimal Word List

The Set Covering algorithm Figure 15 is used to generate the minimal list of words that contains all the in-word tri-phonemes. All the words in the phonetic lexicon are converted into CISAMPA format, with silence (indicated by #) following and preceding them. Then the list searched for the word that increases the tri-phoneme count by the greatest amount. Once a candidate is found it is added to the wordlist and the list is searched again. This process continues till all the 18294 triphonemes are found. The list that is generated contained 4,390 words (for the condition in Figure 15a). However, the major problem with the list is that it consists primarily of words that are long, unfamiliar or borrowed words from other languages as shown in Figure 14. Such a wordlist cannot be effectively used for the ASR system as the native speakers of Urdu will not be able to fluently go through the sentences made from such words. This will prevent the aspects of continuity and spontaneity to be present in the recordings. Besides it would make the job of making the sentences far too hard. However, it must be noted that this is the smallest list for the given corpus that can be generated which covers all tri-phonemes.

ماہرین تابکاریت	نیشنل ایسوسی ایشن فار کار سٹاک ریسنگ
ٹریفک کمیونیکیشن	ہائپر ٹیکسٹ ٹرانسفر پروٹوکول
بےیارومددگار	کسٹمر ریلیشن شپ مینجمنٹ
استادالاساتذہ	بین البینکی شرح منافع
طوفان بادوباراں	درخواست کنندگان

Figure 14 - Example Problematic words in the list

```
; Inputs
;   X is the input corpus
;   C is the condition
; Output
;   O is the output list of words
Greedy-Set-Cover(X, C)
U = X
O = $\phi$
while U $\neq$ $\phi$
     do select word W from U that maximizes C
         U = U - W
         O = O + {W}
endwhile
return O
```

a	`C = |tri-phonemes(W) ∩ tri-phonemes(U)|`

b	`let, N = |tri-phonemes(W) ∩ tri-phonemes(U)|`
	`let, F = Frequency(W)`
	`   then, C = ` w_f `F + ` w_n ` N`

c	`let, N = |tri-phonemes(W) ∩ tri-phonemes(U)|`
	`   let, F = Frequency(W)`
	`   then,`
	`   if N = ` ϕ
	`      C = MIN`
	`   else`
	`      C = F`
	`   endif`

Figure 15 - Modified Minimal Set Cover Algorithm

5.2.3.3 High Frequency Minimal Word List

The solution to the problem presented in the previous section is to give weight to the frequency of occurrence of the words in the corpus as well. Hence the condition of the Set Covering algorithm can be modified, as shown in Figure 15b, to add weights w_f and w_n. This way the weights w_f and w_n can be adjusted to get a minimal list of common (high frequency) words of

Urdu. The list(s) thus obtained contain more words than the one generated in 5.2.3.2 but fulfill the requirements of familiarity of words.

Different values of weights are tried, however the problem of uncommon words continues for most of the experiments. Finally, priority is given to frequency of occurrence of the word in the corpus and subsequent weight to the number of tri-phonemes that it adds to the set (as shown by the condition in Figure 15c). The final wordlist generated contained 11,884 high frequency words.

The major problem with this wordlist is the high number of words. Considering an average sentence length of 8 words, we would end up with 1486 sentences, which are too many to be practically read out by a few speakers. And for the ASR repetition of sentences by multiple speakers is also desired for this data.

5.2.3.4 Reduced high frequency minimal word list

To reduce the size, the wordlist is carefully analyzed and several rules were devised based on the phonetic structure of Urdu and the acoustic properties of the phones. Following are the major rules that are formulated to reduce the size of the in-word tri-phoneme list. The reduction is at the cost of losing some context. However, this is done to so that there is minimal compromise.

5.2.3.4.1 Voiced/voiceless unaspirated stops in context positions

When voiced or voiceless unaspirated stops occur before or after the target phone, their acoustic context has similar effect on the target phone, as long as they have same *place* of articulation [47]. Hence, the affect on the target remains quite minimal (especially spectrally) by the variation in the voicing property (in case of unaspirated stops). An example is shown in Figure 16, in which spectrograms of the three triphones [**t̪ ə b**], [**t̪ ə p**], [**d̪ ə b**] are given ([**d̪**] and [**t̪**] are voiced and unvoiced dental stops, respectively, and, [**b**] and [**p**] are voiced and unvoiced bilabial stops, respectively). It can be seen that the phone [**ə**] changes only slightly by the variation of the voicing property of the context phones, as opposed to change in place. There is pronounced difference in the duration of the vowel, as expected [16].

So, we collapsed all the voiced/voiceless unaspirated stops at the same place occurring at context positions to the voiced version. This reduces the tri-phoneme set significantly. This will however lead to some degree of loss in training but we must remember that the context based phones are being used just to provide enough training as a target occurs with different contexts.

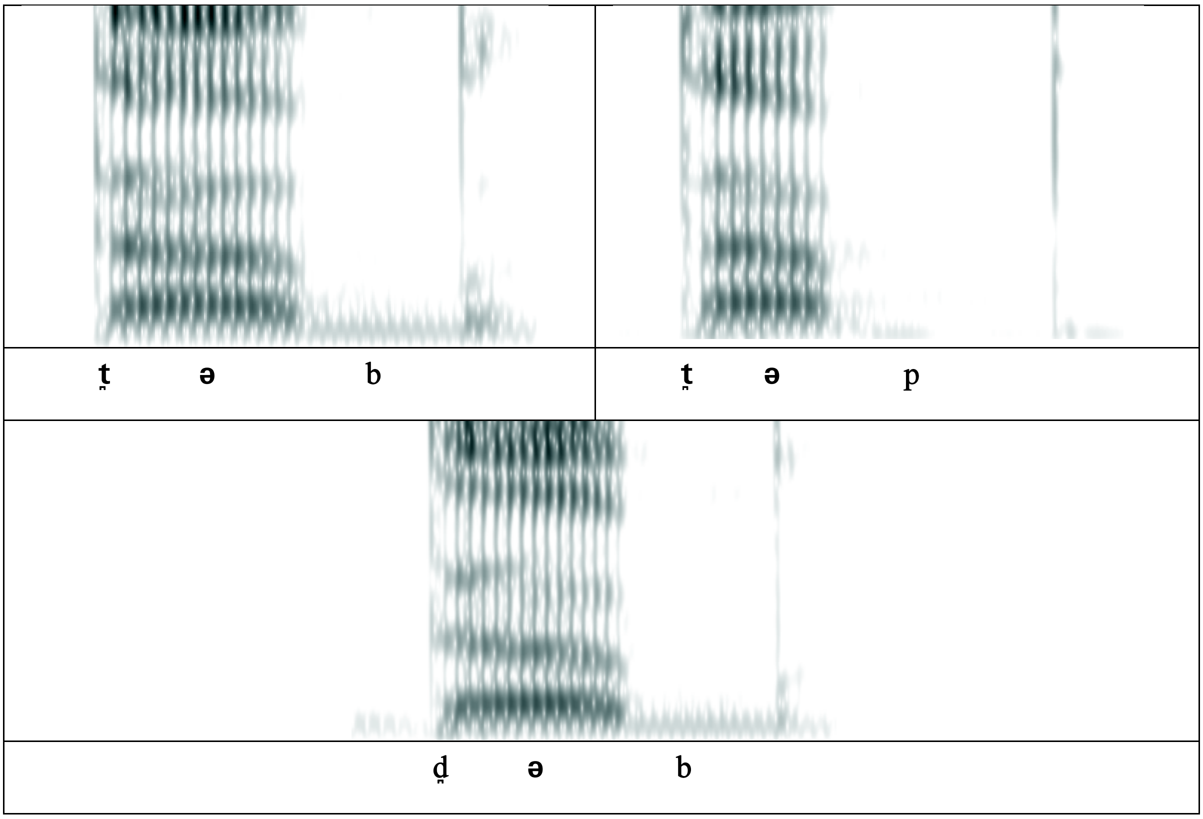


Figure 16 - Acoustic effects of voiced/voiceless unaspirated stops in context positions

5.2.3.4.2 Aspirated/unaspirated stops at tri-phoneme ends

The aspirated/unaspirated stops occurring at the end of tri-phonemes affect the target phone similarly. An example is demonstrated in Figure 17, where triphones [**s ɑ t̪**] and [**s ɑ t̪ʰ**] are compared ([**t̪**] is an unaspirated dental stop, while [**t̪ʰ**] is its aspirated version). It can be seen that the spectrogram is similar, though some breathiness may be introduced towards the end of the vowel. Therefore these two types can be merged (with some compromise).

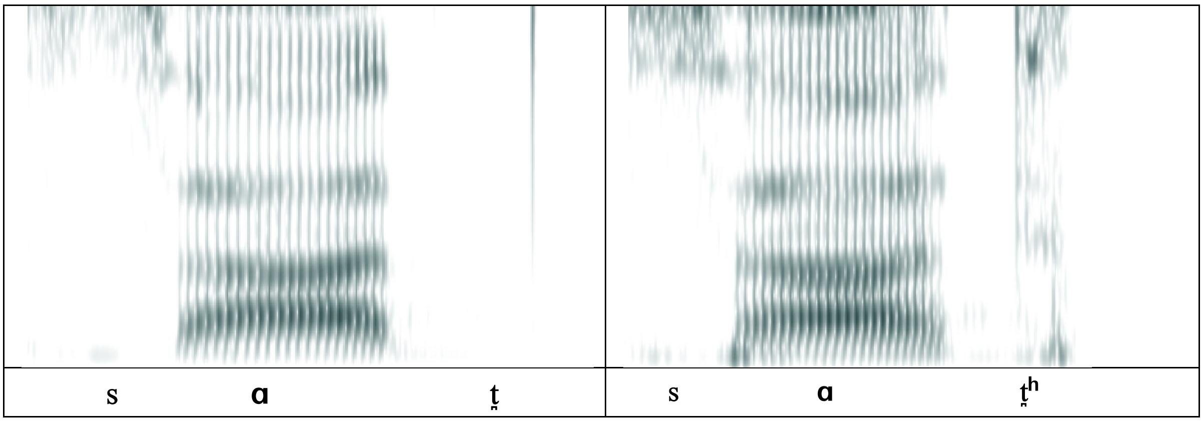


Figure 17 - Effects of Aspiration at triphoneme ends

5.2.3.4.3 Removing low frequency tri-phonemes

Next a frequency analysis of the tri-phoneme list is performed. The goal was to find the frequency of occurrence of each tri-phoneme in the corpus. The resulting frequency list is plotted, as shown in Figure 18. All tri-phonemes occurring more than 10 times are selected for inclusion in the list. At a later stage if there is need in modeling, other tri-phonemes can be added.

As a result of applying the above constraints 9,436 tri-phonemes are removed from the in-word tri-phoneme list, hence leaving behind 8,858 tri-phonemes to cover for recording. Using the algorithm of section 5.2.3.3 the final wordlist generated after removing the tri-phonemes that fall into the above mentioned categories contains 5,681 unique high frequency words. This is comparable with the most optimal list generated earlier using the greedy set cover method which had mostly unfamiliar 4,390 words.

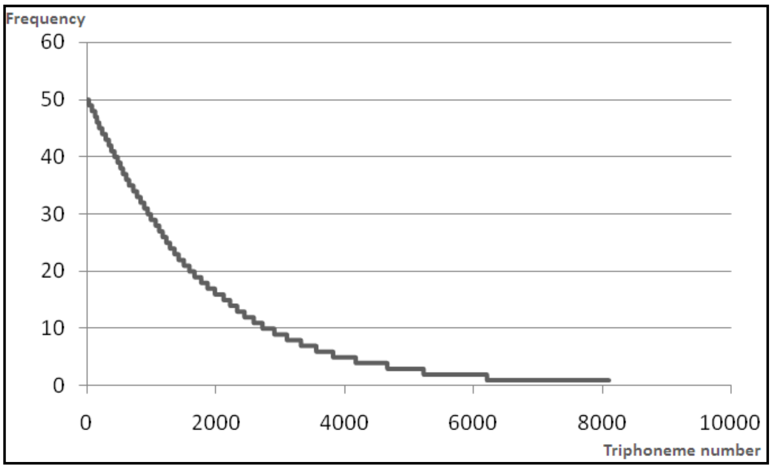


Figure 18 - Plot of tri-phonemes verses frequency in corpus

5.2.4 **Sentence Generation**

The 5681 words generated as described in section 5.2.3, are used by a team of language experts to construct sentences. The aim is to construct sentences that are grammatically correct and sound natural to native Urdu speakers. The guidelines followed during sentence generation are given below:

- Each sentence consists of at least five words
- Sentences with commas are avoided, in order to avoid sentences including lists of items
- Native Urdu speakers should be able to utter the sentence without much difficulty
- The word list has no diacritical marks, so if any words are detected which are ambiguous in pronunciation, sentences are constructed for all variations in the pronunciation, with the appropriate diacritical marks inserted
- Sentences that do not make semantic sense are allowed to be part of the set as long as they are grammatically correct, and easy to read fluently, but should be avoided as much as possible

A total of 708 sentences are produced as a result of this exercise. For quality control, each sentence is reviewed by a member of the team not taking part in the sentence construction. Sentences that are found to be difficult or odd for a native Urdu speaker to utter are identified and sent back to the sentence construction process. For example, Figure 19 shows examples of good and bad sentences. Sentence A is good as it is short, easy to read and makes complete sense. Sentence B is graded as average as it is slightly difficult to read smoothly since the initial part is almost a tongue twister. Some of the words may also be unfamiliar for the average Urdu speaker. Otherwise is correct, grammatically and semantically. Sentence C is only marginally accepted as it is semantically odd, and may cause the reader to react unexpectedly. Grammatically it is correct and over all short. The last example shown as sentence D is rejected because it is too long and difficult. This makes it almost impossible to read through smoothly. Such sentences are reconstructed into other smaller and sensible sentences. The complete list of 708 sentences is shown in the Appendix C.

A	بڑی بحث اور تجزیہ کے بعد یہ فیصلہ ہوا
B	کاشف قزلباش تشنج کے سبب معالجے کیلئے بہترین معالج کے پاس گیا
C	شنید ہے کہ بوبی پنچولی کا حسن نزلے کے سبب مرجھا جانے کو ہے
D	واچ نگر کے کتھک خنازیر شیڈ راؤنڈر میں گوند نچوڑ کراور سوجی چھڑک کر دوروں پر آئے

Figure 19 - Example sentences

5.2.5 Analysis of Sentences and Wordlist

The final sentence list is analyzed and it is found to contain a total of 9804 tri-phonemes and 60 out of 61 phonemes (it misses [l^h], however, the occurrence of this phone is very rare in Urdu [17] and the instances that were found were either rejected as non-words or were removed in the phase where aspiration at word ends was collapsed).

A few words in the wordlist were found to be incorrect and/or lengthy, and are removed from the set by the linguists at the sentence generation phase. The final wordlist from which linguistis generated the sentences was analyzed to confirm that it contained all 8,858 tri-phonemes. The sentences produced by the linguists were then separately analysed and found to contain 9804 triphonemes. The reason for this increase from 8858 to 9804 is that every new word added in the set cover algorithm potentially adds some tri-phonemes which may not already be in the minimal list of triphonemes. Moreover, close class words which may be necessary for making a sentence were also added during the sentence generation phase. The benefit of these additional triphonemes is that they add a reasonable measure of phonetic balance to the sentences.

Figure 21 shows the logarithmic plot of frequency of occurrence of each tri-phoneme in the corpus (the curve above) and its frequency of occurrence in the sentences (shown below). Another feature of interest is the phone-frequency property of the sentences, i.e. the frequency of occurrence of every phone in the sentences. In fact this is the feature which reflects the phonetic balance in the corpus. Figure 20 shows the comparison of the frequencies of occurrence of the Urdu phones in the corpus and the frequencies of occurrence of the same phones in the sentence list. The complete list is shown in Appendix D. This figures (Figure 21 and Figure 20) clearly reflects that phone level (phonetic) balance has been preserved in the sentences (Figure 20), however, triphoneme level phonetic balance is not present (Figure 21). The sentences hence produced cover 60 out of 61 phones of Urdu and are balanced at phone level. They cover all the in-word contextual phones however; are not completely balanced at triphoneme level.

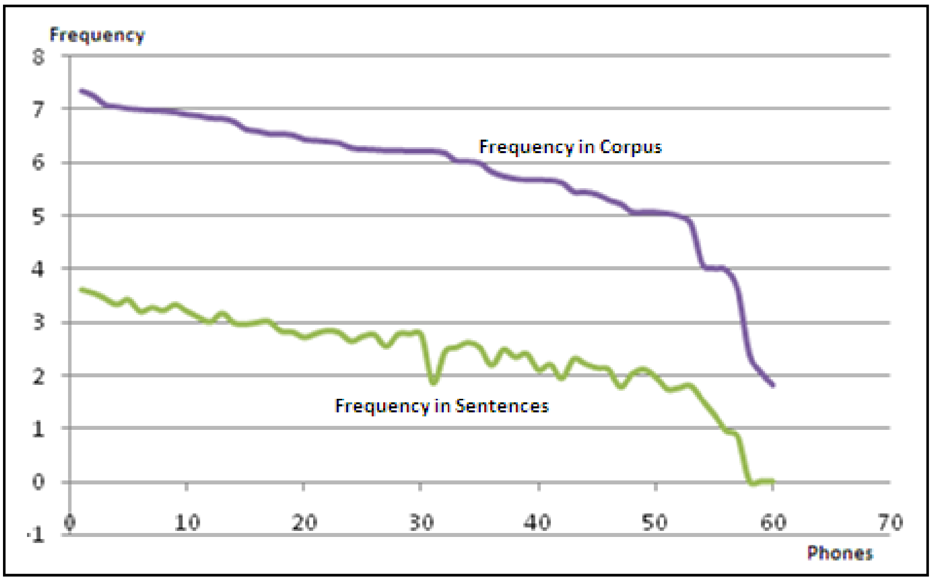


Figure 20 - Phone frequencies ($\log_{10}$) in the corpus vs. the sentences

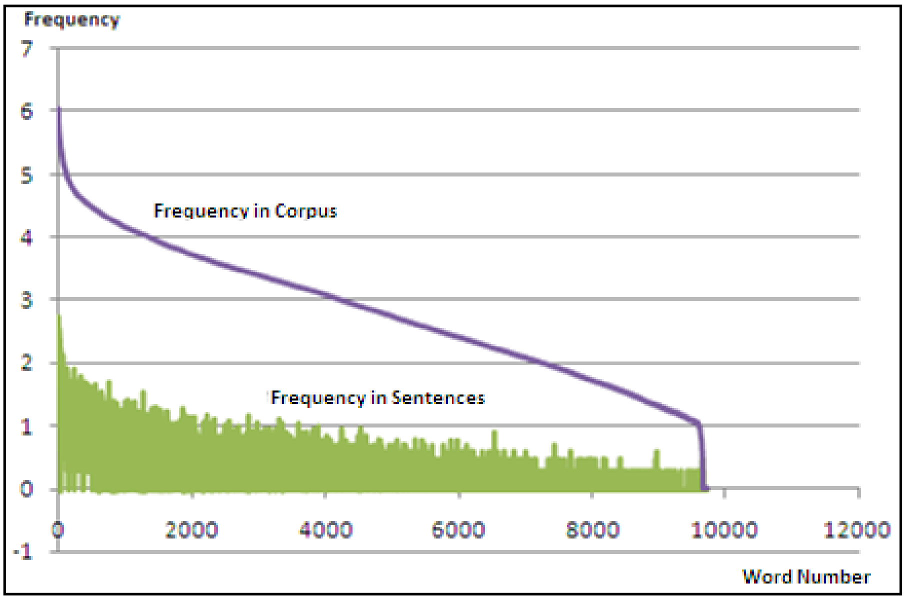


Figure 21 - Triphoneme frequencies ($\log_{10}$) in the corpus vs. the sentences

5.2.6 **Preparation of Interview questions**

The primary goal in the design of interview questions was that the native speakers who are interviewed should be able to respond to them as fluently and spontaneously as possible. Therefore, it comprises of questions related to facts and everyday life. Another motivation behind the design is that the questions should allow for a lengthy answer. The procedure for interviews also caters for the case when the interviewee gets stuck on a question. This is done by asking follow up questions to motivate a detailed response. The questions have been divided into the following two major categories:

5.2.6.1 Factual questions

These serve the dual purpose of gathering specific information about the speaker including their origin, nationality, demography and qualifications related details. Following are the main points about which the speaker is asked:

- Name
- Gender

- Date of birth
- Place of birth
- Area(s) of residence
- Educational institutes attended
- Profession

Not only does this provide the information about the speaker to be used for statistical purposes but also provide useful training data about Proper Nouns (Names of persons, places, institutes) numbers and number formats (dates, years etc.) and names of days of week, months etc. An example interview questionnaire related to this part is shown below (Figure 22):

a. What is your name?
b. What is your gender?
c. What is your height?
d. What is your place of birth?
e. What is your date of birth? Please answer using the format پانچ نومبر انیس سو چالیس
f. Which is your current area of residence?
g. Which, if any, are some other areas of your previous residences?
h. Which school or schools did you attend?
i. Which other educational institutes have you attended, if any?
j. What is your current profession?

Figure 22 - Interview Questions Part-1

5.2.6.2 Natural Speech Elicitation

The goal of this part is to allow the speaker to speak as much and as fluently as possible. The interviewer is required to engage the speaker in a dialog. Each question has a subset of prompts to elicit substantial amount of speech. The purpose of this activity is to induce the volunteer to speak naturally for about fifteen minutes. Completing all the questions is not an objective. The person conducting the session is free to improvise and deviate from the script. This activity should last up to thirty minutes to ensure that at least fifteen minutes of nature speech have been acquired from the speaker. The following figure (Figure 23) shows a sample of questions that can be asked in this session:

a. Explain the route you took to get from your home to this location.
b. How long does it take for you to get to work every day?
c. What are your responsibilities at work?
d. Describe a normal day at work.
e. How did your day go yesterday? Describe with timelines if possible.
f. Describe a memorable day.
g. Describe a funny experience.
h. Describe a scary experience.
i. Describe an interesting experience.
j. What is your greatest fear?
k. Which places would you like to visit and why?
l. Describe an accomplishment that you are proud of.
m. Do you have any brothers or sisters? Are they younger or older than you?
n. Tell us about your friends.
o. Name some of your closest friends.
p. How did you become friends?
q. Tell us about three of your favorite childhood memories.

Figure 23 - Interview questions Part-II

A detailed interview sheet is shown in Appendix E.

5.3 Development of the Speech Corpus

The next phase involved converting the sentence based text corpus into a speech corpus by recording it as a speaker reads it out. Several rules had to be developed to make this process as smooth as possible. The interviews also have to be recorded, split and transcribed. This section provides details and time statistics of this procedure.

5.3.1 **Recording of the Speech Corpus**

The text corpus consists of 708 sentences, some of which are a bit hard to articulate even by a native speaker. This is the result of the greedy approach used in forming the wordlist and sentences. While almost all sentences are grammatically correct, there are some which are semantically confusing. In general the following steps are involved in the speech corpus development:

1. **Directions for recording**
    - The recordings are carried out in normal home or office environment with common ambient background noise.
    - Before starting the session the room is checked for any unique intermittent or continuous noises which may hinder the recording procedure e.g. sounds of crickets and other insects, ticking of clocks, water dripping, repair work etc. in progress inside or outside the building, traffic noise, rain etc. which may not leave a distinct impression upon immediate perception, but will render the recordings unusable.
    - Continuous background noises like the ones emanating from fans, air conditioners etc. may be allowed to continue, if these continue during the entire length of one session
    - It is ascertained that the speaker is not suffering from any major articulatory disorder. For this purpose nasal and voiced vowel sounds uttered by the speaker are recorded and spectrally analysed to ensure correct pronunciation, if necessary
2. **Equipment setup**
    - The recordings are carried out on a laptop computer using an external sound card and microphone to make the recordings laptop unspecific (Creative® External USB sound card and Colorvis® desktop microphone were used for speech data collection). Praat® [48] is used as the main recording and analysis software
    - The recordings are done at a sampling rate of 16000 samples per second, 2 bytes per sample, single channel (mono), using Intel (little Endian byte order) PCM
    - Microphone booster settings are disabled as these make the microphone too sensitive and allows recording low amplitude background noises in a lot of detail

- The microphone is placed in front of the speaker and a little below his/her chin so that the open end of the tube points towards his mouth. A test recording is conducted to ensure that it is not in a horizontal line with his mouth (so that he/she may not breath directly into it)
- Next the microphone is moved a little towards the left or right so that the breath exhaled from the nose does not go directly into it

3. **Testing the setup**
    - The speaker is requested to utter high energy vowels like [ɑ] or [e] (in sentences like: (آ بابا آ) and the recording level is adjusted so that the bar remains within the green zone (in Praat recorder)
    - The recordings are checked to make sure that there is no amplitude clipping
    - Next the speaker is requested to utter strong breathy sounds (like [d̪ʰ], [tʰ], [dʰ]) and the recording is checked to see that he not directly breathing into it (if he is, then the bar will go in the yellow or red zone) and there is no amplitude clipping
4. **Recording Session**
    - The speaker reads out a given number of sentences. It was determined by previous experience that CRULP gained while doing recordings for the Urdu Speech Synthesis system that this number should not exceed 15 in a single recording session, or the quality of utterance suffers (as the speaker may get exhausted).
        a. The 15 sentences are first given to the volunteer to read and get familiar with the pronunciation and become fluent at uttering them; he/she may confirm the diacritics etc. of any confusing word or ambiguities. The IPA transcription is consulted to explain the proper pronunciation of ambiguous/unknown words to the speaker
        b. When the volunteer thinks that he/she is comfortable with the sentences, the recording session begins
        c. The volunteer is required to utter every sentence twice in the following manner:
            i. Utter the sentence with brief pauses after every word. This will not only provide with useful non-continuous speech data, but will also allow the

speaker to gain sufficient fluency as he will utter the sentence in the next phase.

ii. Utter the sentence as fluently as possible, while at a normal and uniform pace.

iii. If the recording person feels that a sentence is uttered wrongly, he/she may ask the volunteer to repeat one or both versions after saying the word "Repeat".

iv. If at any time during the recording there is an overlapping sound like creaking chairs, doors or calls of street hawkers, that part of the recording must be repeated (this requires a lot of alacrity on part of the recording person)

5. **Post-Recording quality check**
   - The sessions are saved in MS WAV format
   - The recorded session is listened to and checked after completion and a repetition is requested wherever required
   - The recording is checked to see that the speaker did not accidently start breathing into the microphone at any time
   - If any disturbing noise overlaps a particular piece of recording, the speaker is requested to repeat that part
   - The global sentence numbers are maintained while recording (the sentence number in the actual list according to which transcription is done) to help in the transcription process
6. **Splitting the recordings**
   - The sessions consisting of 15 sentences, uttered twice, are then split into sentences
   - The sentence files are saved as 16K samples/sec, 16-bit PCM, little Endean PCM, NIST files (for use with Sphinx)
   - Silence is included at the start and end of all the utterances (approximately 100 ms)
   - Files are given names according to global file numbers to help in sound to transcription matching at the time of training

5.3.2 **Timing Statistics of the Recording sessions**

Actual time of recorded speech

These are the actual timing statistics from the recorded data

For 1 hour of recording:

> 1 sentence gives 5 seconds of recorded speech on the average

Therefore,

> 12 sentences will give 1 minute and 720 sentences will give an hour of recording (the total sentences in the list are 708 so that is a benefit). After recording and splitting (and removing repetitions, pauses etc.) it was found that 708 sentences gave 70 minutes of actual speech data

Actual time required for recording 720 sentences (each sentence uttered twice)

1. Average Time required for reading 15 sentences and confirming pronunciation: 5 minutes
2. Average Time required for speaking one sentence twice and correcting errors, mistakes and breaks where these exist by repetition: 1 minute
3. Therefore 15 sentences require 5 minutes for reading + 15 minutes for uttering: 20 minutes
4. 720 sentences (1 hour of recording) require : 960 minutes (16 hours) if spoken continuously without a break and 1200 minutes (20 hours) if a 5 minutes break is allowed after every sentence
5. Note: uttering every sentence only once will save around 15 seconds per sentence i.e. 180 minutes (3 hours) in total but will make the recordings more susceptible to error and during training we may require the sentence with pauses

Actual Time required for splitting 720 sentences (x2)

1. Splitting one session with 15+15 sentences takes on the average 10 minutes
2. Therefore, splitting 720+720 (2 hours of recording) sentences requires: 480 minutes, i.e. 8 hours, which comes to around 4 hours per hour of actual speech data

5.3.3 **Recording of the Interviews**

The guidelines regarding recording setup etc. remain the same as mentioned earlier. The only difference is that the volunteers are informed about the questions to be asked and will be given some time to prepare their responses, if required. They can make brief notes, however, reading out written answers is not allowed.

The interviews are divided into the introduction (factual) and the Natural Speech Elicitation (natural discussion may or may not be based upon facts). The files are recorded, saved and checked as mentioned earlier.

5.3.4 **Splitting and transcribing the interviews**

Splitting interview sessions requires more time than the read speech as it may sometimes be difficult to find a natural pause (that could be mapped on silence) in the discourse. Following rules have been developed by the team of CRULP CMU ASR Project for splitting recorded interviews.

Speech File Segmentation PROCESS

The Speech files are split using *Audacity version 1.2.6* (open source).

The completion of a segment can be marked at:

- The end of a sentence
- The end of an utterance, where the speaker takes a natural pause, this may be due to sentence structure, e.g., a conjunction, or because the speaker is catching his/her breath in preparation of the next utterance

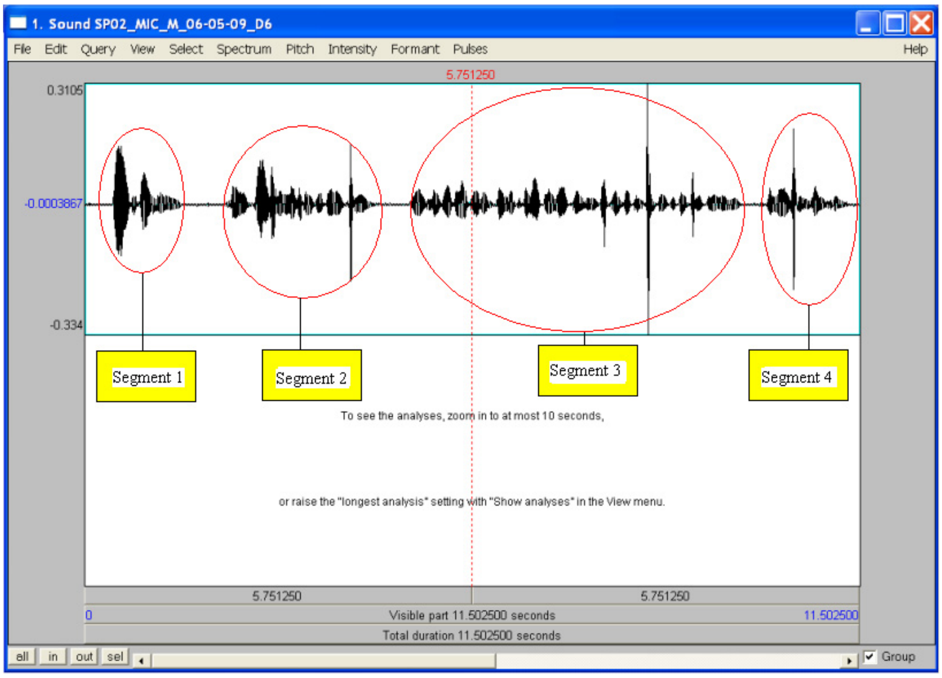
The following is a transcribed sentence showing the pauses and the segments produced from it:

آج کل ہم (vocalized pause) (silence) فاسٹ یونیورسٹی کی طرف سے (vocalized pause) (silence)
سرگودھا کے مختلف دیہاتی علاقوں میں بچوں کو کمپیوٹر کی تربیت دے رہے ہیں(silence) اور میں اس
ٹیم کا حصہ ہوں۔

This sentence will be split into the following four segments:

۱۔ آج کل ہم
۲۔ فاسٹ یونیورسٹی کی طرف سے
۳۔ سرگودھا کے مختلف دیہاتی علاقوں میں بچوں کو کمپیوٹر کی تربیت دے رہے ہیں
۴۔ اور میں اس ٹیم کا حصہ ہوں

The figure below (Figure 24) shows the waveform for the above sentence (in Praat) and marks the different points at which this speech file should be segmented:



To see the analyses, zoom in to at most 10 seconds,

or raise the "longest analysis" setting with "Show analyses" in the View menu.

Figure 24 - Splitting Interview recordings (courtesy of CRULP-CMU ASR team)

5.3.5 Timing Statistics of Interview sessions

The test interviews that were conducted for the thesis showed that the complete questionnaire (shown in Appendix E) took around 2 hours to record and provided with around 90 minutes of speech data. However, the task of splitting and transcription is very time consuming. Following are the actual statistics of the splitting and transcription task for the 120 minutes of interview data by 2 trained linguists:

The total time taken to split interviews into utterances and to transcribe 128 minutes of recorded interviews by trained Urdu typists was around 5460 minutes (91 hours). That would mean that it takes around 42 minutes to split and transcribe 1 minute of spontaneous speech on the average. The 128 minutes of spontaneous speech recordings produced 108 minutes of actual speech after splitting and removing extra silence, noise, repetitions etc.

5.3.6 **Phonetic vs. Phonemic transcription**

This is one of the major questions; *will the speech files be transcribed phonetically or phonemically?* The difference is that in phonemic transcription the words are transcribed as these *should be* uttered. This is mostly rule based and can follow from the Urdu script without even hearing the speech files. This makes it easier to perform and given an Urdu text corpus we can simply generate a phonemic transcription from our (already phonemically transcribed) lexicon. On the other hand the phonetic (or narrow) transcription requires actually hearing the speech files and to perform the transcription as the words have been *actually pronounced.* This is not only a tedious and time consuming process but also brings out further challenges as sometimes the uttered phone can actually lie between the boundary of two phones or map onto a new phone (for that language) altogether. The transcription may require the study of spectrograms to reveal the actual phone designation in addition to perceptual response. However, this approach is better for the acoustic model training as it establishes a more accurate mapping between phones and acoustic waveforms.

For the purposes of the thesis it was decided that the initial training of the speech recognition system should be done on the basis of phonemically transcribed corpora. This will rapidly generate the test results and then on the basis of the error analysis we can phonetically transcribe the corpora in part or as a whole at a later stage in the project. However, diacritics based disambiguation is being done for confusable words of Urdu to facilitate correct lookup of the phonemic lexicon.

5.3.7 **Tools Developed – The Urdu Auto completer and Lexicon Development Utility**

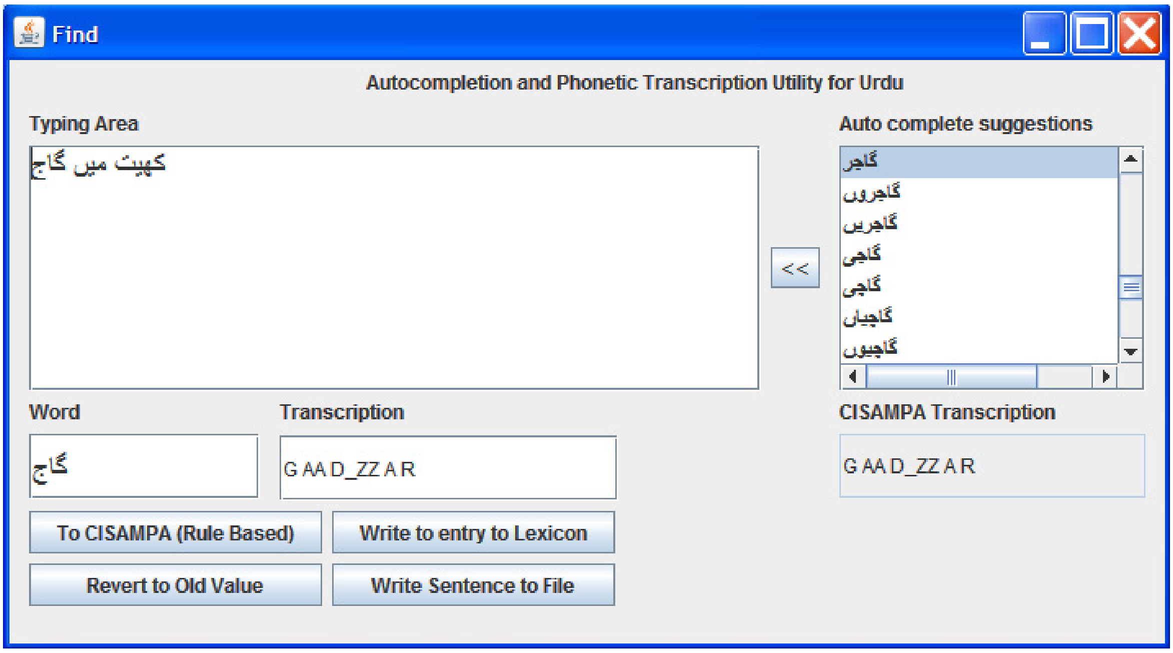
In order to facilitate the task of transcription of the interviews and building of the lexicon an Urdu auto completion and phonetic transcription utility was developed in Java. The main objectives behind the utility were as follows:

- To facilitate the task of Urdu transcription by providing auto complete options from the lexicon
- To prevent spelling errors
- To allow the typist to write the words in exactly the same way as previously available in the lexicon. This prevents errors at a later stage when these transcriptions are compiled for use with Sphinx. This is necessary as many words of Urdu can be correctly written using more than one way (e.g. with or without diacritics, or even with partial diacritics)
- In addition this will prevent or at least reduce collation and normalization errors
- To indicate that a typed word does not exist in the lexicon
- To allow phonetic transcription of words in CISAMPA format
- To allow addition of entries to the lexicon
- To give the facility of rule based letter to sound conversion of Urdu words

Figure 25 shows the interface of the software. The interface and its different uses are summarized below:

- The area indicated as "Typing Area" is the where the sentences to be transcribed are typed. As a word is typed the auto-completion suggestions from the lexicon start appearing in the menu area labeled "Auto complete suggestions"
- This menu can be scrolled using the menu bar or the Page Up and Page Down keys. Pressing Insert key or the button marked "<<" replaced the partially typed word in the Typing Area with the suggested word
- The text area below the auto complete suggestions menu shows the CISAMPA transcription of the suggested word. This way, even if there are no diacritics of the

lexicon entry the typist can recognize the pronunciation of the suggested word and may scroll to choose the most appropriate one


Find
Autocompletion and Phonetic Transcription Utility for Urdu
<<
To CISAMPA (Rule Based)
Write to entry to Lexicon
Revert to Old Value
Write Sentence to File

Typing Area: کھیت میں گاج

Auto complete suggestions:
گاجر
گاجروں
گاجریں
گاجي
گاجی
گاجیاں
گاجیوں

Word: گاج

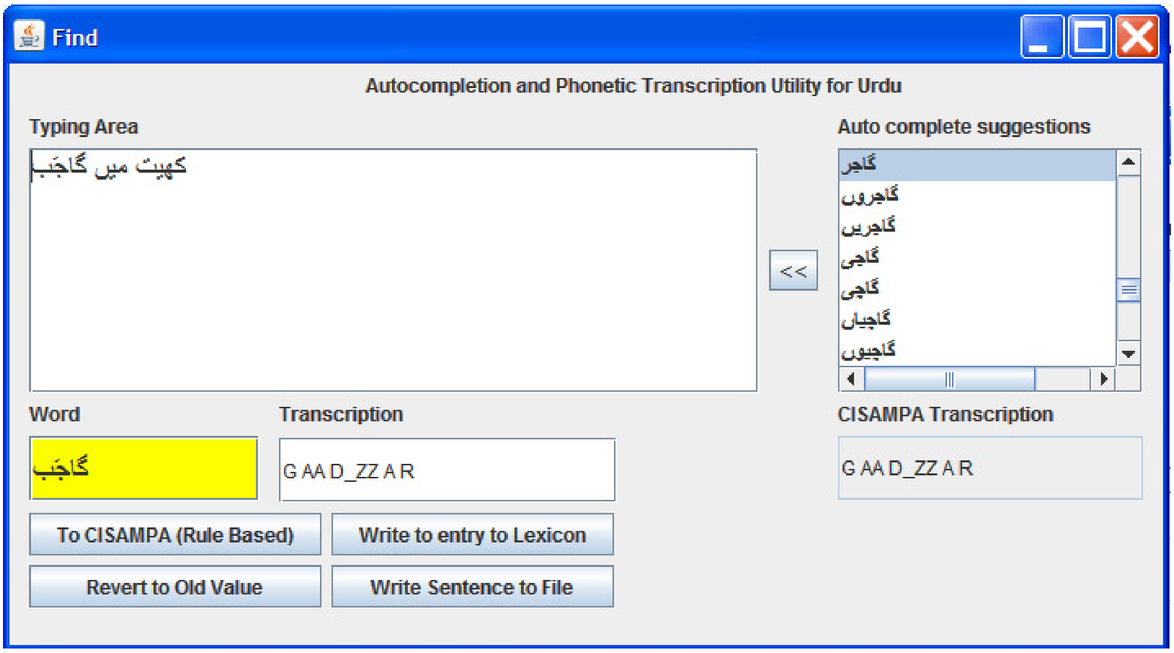
Transcription: G AA D_ZZ A R

CISAMPA Transcription: G AA D_ZZ A R

Figure 25 - Urdu Autocompleter

- The text areas labeled “Word” and “Transcription” are meant to help in the lexicon construction process. That can be accomplished using any of the following two methods:
  - As text is typed in the main Text Area the word being typed also appears in the area labeled “Word” and the CISAMPA transcription of the closest match found in the lexicon is shown in the “Transcription area”. If the word and the transcription is satisfactory at any given time, the typist can press the button “Write Entry to Lexicon” to add the entry and its transcription to the lexicon. This will also add a Romanization of the word to the lexicon as explained in the Section 5.4.1
  - The text area marked “Word” is editable and words can be typed here. Its transcription in the area marked “Transcription” can also be edited and then can be added to the lexicon as explained in the previous section

- If a new word is typed in the Main Text Area i.e. a word not previously in the lexicon, the text area marked "Word" turns yellow to indicate this as shown in Figure 26. The typist can simply add the transcription in the transcription area by hand using the lexical best match suggestion (which is already present there) or can use the Letter to Sound option as explained below



Typing Area: کھیت میں گاجَب

Auto complete suggestions:
گاجر
گاجروں
گاجریں
گاجی
گاجی
گاجیاں
گاجیوں

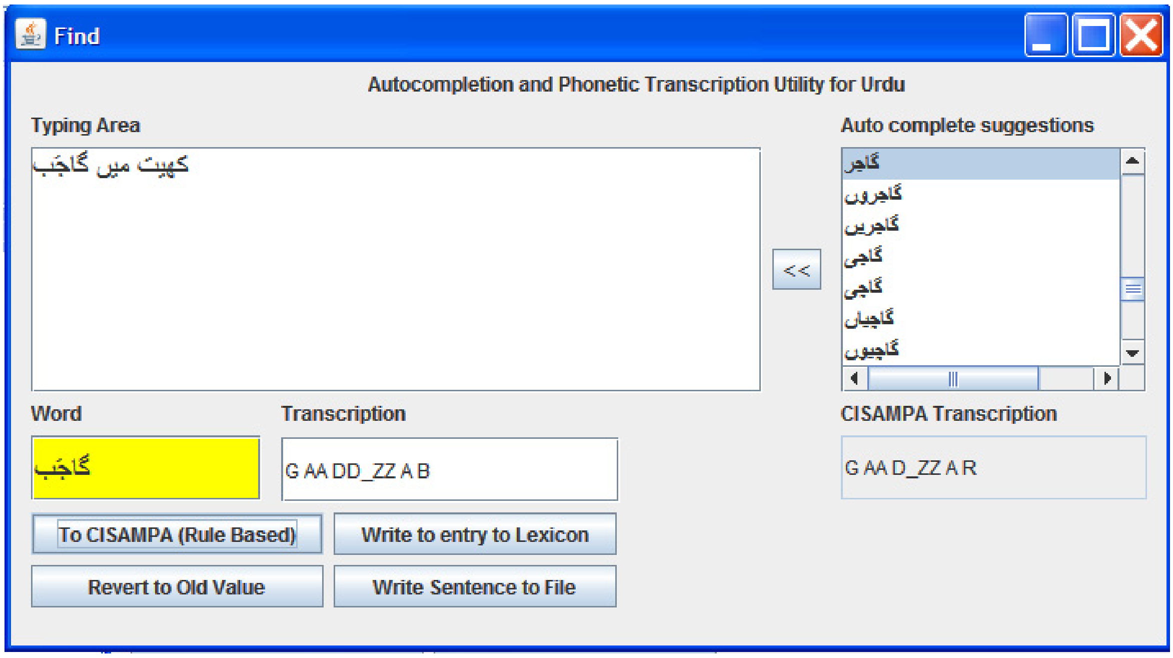
Word: گاجَب

Transcription: G AA D_ZZ A R

CISAMPA Transcription: G AA D_ZZ A R

Figure 26 - Auto completer new word warning

- The "To CISAMPA (Rule Based)" option provides letter to sound conversion option for the text written in the Word text area. Pressing this button gives the CISAMPA transcription suggestion in the transcription text area which can be modified if required. As shown in Figure 27, the suggestion gives the correct transcription for this hypothetical word as it is completely diacritized, otherwise it would have given partial transcription



Typing Area: کھیت میں گاجَب

Auto complete suggestions:
گاجر
گاجروں
گاجریں
گاجی
گاجی
گاجیاں
گاجیوں

Word: گاجَب

Transcription: G AA DD_ZZ A B

CISAMPA Transcription: G AA D_ZZ A R

Figure 27 - The letter to sound option

- Finally the button marked "Write Sentence to File" writes the sentence to a predefined Unicode file. The button "Revert to Old Value" simply revert from the letter to sound rules based CISAMPA suggestion to the lexical suggestion in case if it is ever desired

5.4 Compilation of files required by Sphinx

The Sphinx speech recognition system requires many different files in specific formats to be able to perform the training and decoding tasks. Manually generating these files is a lengthy job which is also more susceptible to errors, which may not be easy to detect at a later stage. For this project the Sphinx-3 trainer and Sphinx-3 decoder has been used. Some tests were also performed with Sphinx-4. This section discusses the files required for all these systems as well as the Sphinx file compiler that I developed to facilitate the file generation and data analysis tasks. Let us start with some basic issues and their solutions that we had to come up with.

5.4.1 Phonetic transcription using CISAMPA

The phonetic transcription (actually phonemic as mentioned earlier) needs to be done is some phonetic notation. IPA uses the Unicode character set and is hence not usable as Sphinx does not as yet recognize Unicode. This led us to use the SAMPA character set which we had to abandon very soon as Sphinx is not case sensitive and hence does not differentiate between capital and small case letter; whereas SAMPA distinguishes between many characters on the basis of case like n (for [n]) vs N (for [ŋ]). We could not use X-SAMPA [49] as it largely relies upon special characters in its character set e.g. \, < etc. Since these characters needed to be used as files names as well (where use of many special characters is not allowed e.g. \) and moreover certain software systems treat these special characters as control characters or position markers. Therefore, a notation free from special characters was required. ARPABET [50] could have provided the solution but the ARPABET notation is too specific for American English pronunciation (for which it has been developed) and is difficult to read for Urdu sounds. E.g. Urdu word بجلی ([b ɪ ʤ l i]) is more readable as B I D_ZZ L II (in CISAMPA) than B IH JH L IY (in ARPABET) or بڑا ([b ∂ ɽ ∂]) can be represented as B A RR A (in CISAMPA) but I was unable to find any character for the retroflex [ɽ] in ARPABET, same goes for many other Urdu specific symbols like nasal vowels. In short, ARPABET is too English specific and not suitable for Urdu.

Therefore a notation similar to SAMPA was needed which is also case insensitive. Hence using SAMPA character set as a starting point a phonetic character set was developed for the purposes of this thesis and project. It was named CISAMPA (Case Insensitive SAMPA). The character sets are shown side by side in Appendix A. The basic rules of conversion from SAMPA to CISAMPA are as follows:

- The complete character set is in capital case
- The character set does not include any punctuation mark or special character like @ or / or ? etc. as these may be treated as control characters or position markers in different software systems
- Most of the consonants have been converted simply by converting them into capital form

- Dentals are indicated by an _D, as [t̪] is represented as T_D
- Aspiration is indicated by _H, as [t̪ ʰ] is represented by T_D_H
- Retroflex is indicated by double characters e.g. [ʈ], [ɖ] and [ɽ] are represented as TT, DD and RR (this remains true for the alveolar versions of the former two as well i.e [t] and [d])
- Short vowels are indicated by single capital character while long ones by double capital characters (A for [∂] and AA for [ɑ])
- Some vowels are represented by ARPABET like notations like [e] is represented as AE
- Nasals are represented by appending an N e.g. [ẽ] is represented as AEN

5.4.2 **Unicode text format**

Sphinx does not support Unicode text format as yet, while Urdu script uses Unicode characters. Therefore, a Unicode to ASCII mapping mechanism was required. A simple solution for this problem was developed. As the phonemic transcription in CISAMPA is done using the lexical lookup, and the CISAMPA notation is completely ASCII based therefore the Romanization is simply done by removing the spaces from the CISAMPA transcription. This produces a one-to-many mapping between Urdu and CISAMPA but a one-to-one mapping between Romanization and CISAMPA. The phonetic lexicon hence contains entries in Urdu mapped to Romanization and CISAMPA transcription as shown in Figure 28 (extra tabs have been added to improve readability) which is a sample from the phonetic lexicon.

Urdu	CISAMPA	Romanization
أسي	A S S II	ASSII
أن	A N	AN
أجاگر	U D_ZZ AA G A R	UD_ZZAAGAR
أدھر	U D_D_H A R	UD_D_HAR
أس	UU S	UUS
أسے	U S AE	USAE
أميد	U M II D_D	UMIID_D
أٹھا	U TT_H AA	UTT_HAA
أٹھاؤں	U TT_H AA UUN	UTT_HAAUUN

Figure 28 - Phonetic Lexicon

5.4.3 **Files required by Sphinx for training**

Following files are required by Sphinx for training:

5.4.3.1 Audio Data

The audio data files (speech files) are required to be in NIST or .WAV format. These files will be used in developing the Mel Frequency Cepstral Coefficients (MFCCs) using which, the system will be trained. The files I used were in .nist format, single channel, sampled at 16000 Samples per second at 16 bits per sample in little Endean format. All utterances (sentences) were followed and preceded by silence.

5.4.3.2 Dictionary File

The dictionary file establishes the word to phonetic transcription mappings. In our case it contains Roman to CISAMPA transcription mapping. More than one pronunciation mappings can be shown with a (1) and (2) etc. after the word.

For example

AEK	AE K
AEK(1)	AA AE K

However, in our case as the Romanization is done after CISAMPA transcription there will only be a one-to-one mapping. The one-to-many mapping can be done in the phonetic lexicon between Urdu and CISAMPA transcription. A sample from the dictionary file is shown in Figure 29.

```
LAAUBAALII          L AA U B AA L II
D_DUGNAA            D_D U G N AA
SHAAGIRD_D          SH AA G I R D_D
T_SHARR_HAAOO       T_SH A RR_H AA OO
LAAD_DIINJAT_D      L AA D_D II N J A T_D
ATTTT_HAAIIS        A TT TT_H AA II S
T_DAD_ZZARBAA       T_D A D_ZZ A R B AA
PUUNT_SH_HNAE       P UU N T_SH_H N AE
D_ZZONAA            D_ZZ O N AA
KAYOOLARII          K AY OO L A R II
BILAAXOF            B I L AA X O F
```

Figure 29 - Dictionary File

5.4.3.3 Filler Dictionary File

The filler dictionary contains the filler words e.g. the words for mentioning silence and special sounds like cough, breath, chair creaking, door closing etc. I have defined the non-speech utterances i.e. the start of utterance silence <s>, the end of utterance silence <\s> and the middle of utterance silence <sil> in the filler file. I have mapped them all to the same phone SIL, which models silence or the background noise. A sample is shown in Figure 30.

```
<s>             SIL
</s>            SIL
<sil>           SIL
```

Figure 30 - Filler File

5.4.3.4 Phone Definition File

The phone file defines all the phones used in the dictionary file including the silence SIL There should be no empty lines in this file. A sample is shown in Figure 31.

```
G_H
Z
Y
AE
X
DD
V
AA
U
S
AEN
R
Q
SH
SIL
```

Figure 31 - Phone File

5.4.3.5 File IDs definition file

In the file train file ids file, all audio file ids without extensions with references to the root folder are defined. A sample is shown in Figure 32

```
an4_train/c1
an4_train/c2
an4_train/c3
an4_train/c4
an4_train/c5
an4_train/c6
an4_train/c7
an4_train/c8
an4_train/c9
an4_train/c10
an4_train/c11
an4_train/c12
```

Figure 32 - File IDs

5.4.3.6 Transcription File

The train transcription file establishes sentence to utterance mappings. These are not phone to audio mappings but mappings between the words in the left column of the dictionary file and the audio files. It is important to note is that the files should be in the same order as described in the training file IDs file. First few lines of my file are shown in Figure 33.

```
<s>   NIILAM NAE SAALGIRAA PAR HAYDD SAYSMOOGIRAAF ASVAD_D QURAYSHII KAE
MAAT_D_HAE PAR AYNTT_HAN OR 7AM KII AAT_DISHIIN RO MEHSUUS KII </s> (c1)
<s>   HAAD_ZZII MUD_ZZAAHID_D BILGIRAAMII MAXZAN OR 7AZVAA KAE AEK ARAB
QAARAIIN MAEN INT_DIHAAII SAAD_DIQ OR D_ZZAYNUUIN QAARII T_D_HAE </s> (c2)
<s>   SAAMAEIIN INFAARMAESHAN KII G_HAN GARAD_ZZ SUNAEN T_DOO VIIZAE KII
RIPOORTT MAEN POOSHIID_DAA AEK MEHD_DUUD_D EL VII DDOOMAYSTTIK PAYKID_ZZ
HAE </s> (c3)
<s>  KAMJOONISTT LOOGOON NAE T_DANG HOONAE KAE BAAVAD_ZZUUD_D KAII SHUBOON
MAEN T_DAND_DIHII SAE APNAE KAYRIIAR KOO MUZAYJAN KAR LIJAA </s> (c4)
<s>    TTARAANSFAARMAR  PAR  MIDDNAAITT  MAEN  SHAAHIIN  GID_D_H  OR  UQAAB
SAMAET_D T_SHESTT KAE BAL SARAEAAM SAYNKRROON BARDD BAYTT_HT_DAE HAYN </s>
(c5)
<s>  KARRVAE QAHVAE KAA SHAYD_DAAII AS7AR KAASHMIIRII BAA7BAANII SIIK_HNAE
VAALAA PAANT_SHVAAN MANAED_ZZAR HAE </s> (c6)
<s>    PATT_HTT_HOON  OR  SIRKAA  HAAD_DSAAT_DII  D_DARD_D  BAAM  SAE  FORAN
SULD_ZZ_HAAAEN P_HIR MIITT_HAE OR KOOLAYSTTAROOL SAE BAT_SHAEN </s> (c7)
```

Figure 33 - Transcription File

5.4.4 **File required by Sphinx-3 for decoding and running tests**

To recognize given speech files, the Sphinx-3 decoder can use the acoustic model files that are generated in the training phase and a language model file, generated separately. The tests can also be run in batch mode to calculate Word Error Rate, Accuracy and to get detailed alignment results. Following are the main files required for running the batch mode tests:

5.4.4.1 Test Audio speech files

Just like the training mode, the files to be recognized should be present in NIST or WAV format in the given directories. The file characteristics can be set in the decode configurations file.

5.4.4.2 Test File IDs

Like the Train File IDs, this file contains the file names of the test files, placed in the root test directory (can be defined in the decode configurations file).

5.4.4.3 Test Transcription

This file contains the names and word transcription of the test files as the train transcriptions file. The files are mentioned in the same order as in the test file IDs file. These are considered to be the reference transcriptions against which Sphinx compares its decoding hypothesis, aligns these two and calculates the error and accuracy results.

5.4.4.4 Language Model

The Unigram, Bigram or Trigram language models can be generated using either the online language modeling utilities [51] if the unique word count is less than 5000, or the Statistical Language Modeling (SLM) toolkit [14], which works in Linux environment. The SLM toolkit is also useful as it allows more options for generating the Language Models, e.g. different smoothing strategies like *absolute*, *Witten-Bell discounting*, *Linear Interpolation* and *Good-Turing Smoothing*. There are many other options described in Appendix G. The ASCII based language model has to be converted into a binary dump file using the utility lm2dmp, as Sphinx-3 recognizes binary language model files. Figure 34 shows a few lines from a Linear Interpolation based trigram language model generated by the SLM toolkit.

```
###########################################################################
## Copyright (c) 1996, Carnegie Mellon University, Cambridge University,
## Ronald Rosenfeld and Philip Clarkson
###########################################################################
===========================================================================
===
============  This file was produced by the CMU-Cambridge   ===============
============     Statistical Language Modeling Toolkit      ===============
===========================================================================
This is a 3-gram language model, based on a vocabulary of 5658 words,
  which begins "7AAFIL", "7AAIB", "7AAJAT_D"...
This is an OPEN-vocabulary model (type 2)
  (0.50 of the unigram discount mass was allocated to OOVs)
Linear discounting was applied.
1-gram discounting ratio : 0.557447
2-gram discounting ratio : 0.166305
3-gram discounting ratio : 0.0673904
This file is in the ARPA-standard format introduced by Doug Paul.

p(wd3|wd1,wd2)= if(trigram exists)            p_3(wd1,wd2,wd3)
                else if(bigram w1,w2 exists) bo_wt_2(w1,w2)*p(wd3|wd2)
                else                          p(wd3|w2)

p(wd2|wd1)= if(bigram exists) p_2(wd1,wd2)
            else              bo_wt_1(wd1)*p_1(wd2)

All probs and back-off weights (bo_wt) are given in log10 form.

Data formats:

Beginning of data mark: \data\
ngram 1=nr             # number of 1-grams
ngram 2=nr             # number of 2-grams
ngram 3=nr            # number of 3-grams

\1-grams:
p_1     wd_1 bo_wt_1
\2-grams:
p_2     wd_1 wd_2 bo_wt_2
\3-grams:
p_3     wd_1 wd_2 wd_3

end of data mark: \end\

\data\
ngram 1=5659
ngram 2=9871
ngram 3=10881

\1-grams:
-0.3541 <UNK>  0.0000
-4.3151 7AAFIL -0.0790
-4.3151 7AAIB  -0.0790
-4.3151 7AAJAT_D -0.0790
-4.3151 7AALIB -0.0750
-4.3151 7AAN   -0.0789
-4.3151 7ABAN  -0.0790
-4.3151 7AD_DD_DAAR-0.0790
-4.3151 7AFLAT_D -0.0732
-4.3151 7AFUUR -0.0686
-4.3151 7AJJUUR  -0.0790
-4.3151 7ALAT_D  -0.0750
-4.3151 7AM -0.0707
-4.3151 7ANII  -0.0790
-4.3151 7ARIIB -0.0719
-4.3151 7ASAB  -0.0785
-4.3151 7ASHII -0.0732
…
-2.9523 KOO HUVAYD_DAA 0.0487
-2.9523 KOO IIFAA 0.0487
-2.9523 KOO JAELOO 0.0487
-2.9523 KOO KAAMJAAB 0.0074
-2.9523 KOO KALIID_DII 0.0487
-2.9523 KOO KANVAENS 0.0487
-2.9523 KOO KAT_SHOOKAE 0.0487
-2.6513 KOO KIJAA -0.0271
-2.9523 KOO KIRAYSH 0.0487
-2.9523 KOO KOOHIST_DAAN 0.0487
-2.9523 KOO K_HIRRKII 0.0487
-2.9523 KOO K_HULAE 0.0074
-2.9523 KOO LAAHIQ 0.0487
…
-1.1714 ZUAMAA OR MUAZZIZIIN
-1.1714 ZUBAAN GAMB_HIIR NAA
```

```
-1.1714 ZUHR T_DAK SAANSAEN
-1.1714 ZUHUUR XAAN KAE
-1.1714 ZULAYXAA SAHAELII KII
-1.1714 ZULD_ZZALAAL D_DAAOO OR
-1.1714 ZULD_ZZINAA KII SUFAYD_DII
-1.1714 ZULFAEN T_DOOLJAE SAE
-1.1714 ZULFII PARAAIIVAETT HOOMIIOOPAYT_D_HII
-1.1714 ZULFIQAAR KII KAII
-1.1714 ZULFOON KAE SINNGG_HAAR
-1.1714 ZULM KAII GANNAA
-1.1714 ZUMRAE MAEN AAT_DII
-1.1714 ZURUURIIJAAT_D ZIJAAD_DAA AZAAB
-1.1714 ZUUD_D KOSH AEZAAZ
-1.1714 ZUUD_DHIS OR OOT_SH_HAE
-1.1714 ZZANG D_ZZAYGUUAR MASHIINRII
```

Figure 34 - Language Model File

5.4.5 **Test Files required for Sphinx-4 decoder**

The Java based Sphinx-4 decoder requires the following files for running the performance tests:

5.4.5.1 Speech Data Files

The speech files should be in header less raw data format (.raw), in single channel, 16K samples/second, 16 bit Big Endean Format. The SOX audio converter for windows [52] was used to convert the NIST files in batch mode into RAW files.

5.4.5.2 Wordlist File

It contains the wordlist grammar. We can also run tests on Bigram and Trigram grammar files. Figure 35 shows an example.

```
T_DAHSIILOON
HAD_DD_DAENAZAR
RANGVAAAEN
HAARVAYSTT
SAARIFIIJAT_D
PARSAAD_D
P_HISALT_DII
T_SHAAILDDLAEBAR
KAB_HUU
D_ZZUUT_DIJAAN
SAMAD_ZZ_HNAE
T_D_HIRII
SAMAD_ZZ_HNAA
```

```
P_HUULAEN
SUD_D_HARNAE
SUD_D_HAART_DII
FAYKSIZ
MUUDD
D_DILGURD_DAE
BAYLT_SHAHBARD_DAAR
VAD_ZZD_DAAFRIIN
HAYD_ZZAAN
TTAYLIIVIIZZANSIKRIIN
```

Figure 35 - Wordlist grammar file

5.4.5.3 Transcription Batch File

This is the main test file where file names, file paths and transcription of the sentences is given. This transcription is used as the test reference. Figure 36 shows an example.

```
/an4/wav/c1.raw NIILAM NAE SAALGIRAA PAR HAYDD SAYSMOOGIRAAF ASVAD_D
QURAYSHII KAE MAAT_D_HAE PAR AYNTT_HAN OR 7AM KII AAT_DISHIIN RO MEHSUUS
KII
/an4/wav/c2.raw HAAD_ZZII MUD_ZZAAHID_D BILGIRAAMII MAXZAN OR 7AZVAA KAE
AEK ARAB QAARAIIN MAEN INT_DIHAAII SAAD_DIQ OR D_ZZAYNUUIN QAARII T_D_HAE
/an4/wav/c3.raw SAAMAEIIN INFAARMAESHAN KII G_HAN GARAD_ZZ SUNAEN T_DOO
VIIZAE KII RIPOORTT MAEN POOSHIID_DAA AEK MEHD_DUUD_D EL VII DDOOMAYSTTIK
PAYKID_ZZ HAE
/an4/wav/c4.raw KAMJOONISTT LOOGOON NAE T_DANG HOONAE KAE BAAVAD_ZZUUD_D
KAII SHUBOON MAEN T_DAND_DIHII SAE APNAE KAYRIIAR KOO MUZAYJAN KAR LIJAA
/an4/wav/c5.raw TTARAANSFAARMAR PAR MIDDNAAITT MAEN SHAAHIIN GID_D_H OR
UQAAB SAMAET_D T_SHESTT KAE BAL SARAEAAM SAYNKRROON BARDD BAYTT_HT_DAE
HAYN
/an4/wav/c6.raw KARRVAE QAHVAE KAA SHAYD_DAAII AS7AR KAASHMIIRII
BAA7BAANII SIIK_HNAE VAALAA PAANT_SHVAAN MANAED_ZZAR HAE
/an4/wav/c7.raw PATT_HTT_HOON OR SIRKAA HAAD_DSAAT_DII D_DARD_D BAAM SAE
FORAN SULD_ZZ_HAAAEN P_HIR MIITT_HAE OR KOOLAYSTTAROOL SAE BAT_SHAEN
/an4/wav/c8.raw KAYOOLARII DDAYM KII MOZUUNIIJAT_D SAMD_ZZ_HAANAE OR
ID_DAARAE KAE ARKAAN KII T_DARBIIJAT_DII XID_DMAAT_D KAE LIIAE HABIIB NAE
T_DIHRIIK T_SHALAAII
/an4/wav/c9.raw IQT_DISAAD_DII MUAAMALAAT_D KII T_DAFHIIM OR FARMAAN
BURD_DAAR NOOD_ZZAVAANOON KII BARIIFING GUFT_DGUU MAEN SARAEFAHRIST_D
LIK_HAEN
/an4/wav/c10.raw SAD_ZZD_ZZAAD_D RAYSKIJUU MALUUMAAT_D K_HANGGAALNAE KAE
BAAD_D D_ZZAANAE KIJUUN BILAASOOT_SHAE AT_SHAANAK SUUIMING PAR RAA7IB HAE
```

Figure 36 - Sphinx-4 transcription file

5.4.6 **Tools developed – the Sphinx Compiler**

As can be seen from the above given descriptions, if all these files are generated manually, it will be a very time consuming task and will result in a lot of errors. Therefore, it was required to automate this process as much as possible. A Java based program was developed for the generation of these files and for the analysis of training and test data as well. The software also provides many utility functions e.g. CISAMPA based Roman to Urdu Unicode converter etc. that are essential for the completion of decode process back to Urdu. Following sections give a detailed description of the files required as input along with the file format and examples. And sample output files.

5.4.6.1 Description of Input Files

Following is a brief description of the input files required by different functions of the Sphinx Files Compiler:

5.4.6.1.1 Unicode Training data File

This is a simple sentence corpus of Urdu, which provides the transcription of the audio data files used for training. The file is required to contain one sentence per line (in the same order as the speech files in which these sentences are uttered). The file must be in 16 bit Unicode, Big Endean Format. Figure 37 shows a sample of first few lines.

```
نیلم نے سالگرہ پر ہیڈ سیسموگراف اسود قریشی کے ماتھے پر اینٹھن اور غم کی آتشیں رو محسوس کی
حاجی مجاہد بلگرامی مخزن اور غزوہ کے ایک ارب قارئین میں انتہائی صادق اور جینوئن قاری تھے
سامعین انفارمیشن کی گھن گرج سنیں تو ویزے کی رپورٹ میں پوشیدہ ایک محدود ایل وی
ڈومیسٹک پیکج ہے
کمیونسٹ لوگوں نے تنگ ہونے کے باوجود کئی شعبوں میں تندہی سے اپنے کیریئر کو مزین کر لیا
ٹرانسفارمر پر مڈنائٹ میں شاہین گدھ اور عقاب سمیت چیسٹ کے بل سرعام سینکڑوں برڈ بیٹھتے
ہیں
کڑوے قہوے کا شیدائی اصغر کاشمیری باغبانی سیکھنے والا پانچواں منیجر ہے
```

Figure 37 - Unicode Training Data File

5.4.6.1.2 Unicode Test data File

This is a sentence based corpus for the test data. The requirements and format are the same as for the training data file shown in section 5.4.6.1.1.

5.4.6.1.3 Phonetic Lexicon

The phonetic lexicon (as shown in Figure 28) is also in 16 bit Unicode Big Endean format. It presents a mapping between Urdu words, CISAMPA transcription and Romanized versions of those words. If a word has multiple pronunciations it must be defined with as many entries with different CISAMPA and Romanization in each case. Thus this file produces a one-to-many mapping from Urdu to CISAMPA transcription and Romanization.

5.4.6.1.4 Decode Results File

Sphinx-3 and Sphinx-4 produce the decoding results in the given Romanization format. This file contains the ordered decoding results. It is in plain ASCII file format as shown below in Figure 38.

```
NIILAM NAE SAALGIRAA PAR HAYDD SAYSMOOGIRAAF ASVAD_D QURAYSHII KAE
MAAT_D_HAE PAR AYNTT_HAN OR 7AM KII AAT_DISHIIN RO MEHSUUS KII (ct1)
HAAD_ZZII MUD_ZZAAHID_D BILGIRAAMII MAXZAN OR 7AZVAA KAE AEK ARAB QAARAIIN
MAEN INT_DIHAAII SAAD_DIQ OR D_ZZAYNUUIN QAARII T_D_HAE (ct2)
SAAMAEIIN INFAARMAESHAN KII G_HAN GARAD_ZZ SUNAEN T_DOO VIIZAE KII
RIPOORTT MAEN POOSHIID_DAA AEK MEHD_DUUD_D EL VII DDOOMAYSTTIK PAYKID_ZZ
HAE (ct3)
KAMJOONISTT LOOGOON NAE T_DANG HOONAE KAE BAAVAD_ZZUUD_D KAII SHUBOON MAEN
T_DAND_DIHII SAE APNAE KAYRIIAR KOO MUZAYJAN KAR LIJAA (ct4)
TTARAANSFAARMAR PAR MIDDNAAITT MAEN SHAAHIIN GID_D_H OR UQAAB SAMAET_D
T_SHESTT KAE BAL SARAEAAM SAYNTTAR OO BARDD BAYTT_HT_DAE HAYN (ct5)
KARRVAE QAHVAE KAA SHAYD_DAAII AS7AR KAASHMIIRII BAA7BAANII SIIK_HNAE
VAALAA PAANT_SHVAAN MANAED_ZZAR HAE (ct6)
```

Figure 38 - Sphinx-3 results file

5.4.6.2 Functions provided by Sphinx Compiler

5.4.6.2.1 Construct Dictionary File

Inputs: Unicode Training Data File, Unicode Test Data File, Phonetic Lexicon.

Output: Dictionary file for Sphinx in the required format, with Roman Urdu to CISAMPA transcription mappings.

Options: Three options are available:

1. Generate the dictionary only from the words in the Unicode Training data file
2. Generate the dictionary from the combined set of test and training data
3. Generate the dictionary from all the entries in the phonetic lexicon which are composed entirely of the phones which were present in the training data

Errors: The function will exit with error if a word is encountered which is not available in the Phonetic lexicon (Please see section 5.4.6.2.9.1).

5.4.6.2.2 Construct Romanized Corpus for Language Model Generation

The corpus for Language Model generation is required to be in ASCII encoding. This is to be supplied as input to the online Language Model tool or the Statistical Language modeling toolkit.

Inputs: Unicode Training Data File, Unicode Test Data File, Phonetic Lexicon

Output: Romanized version of the Input Corpora

Options: Four options are available:

1. Generate the Corpus only from the training data without silence markers (<s> and </s>) before and after each sentence (this can be used with the online Language Modeling toolkit which can insert these silence markers itself)
2. Generate the Corpus only from the training data with silence markers (start: <s> and end: </s>) before and after each sentence (This is required by the SLM toolkit)
3. Generate the Corpus from the combined training and test data without silence markers (<s> and </s>) before and after each sentence (this can be used with the online Language Modeling toolkit which can insert these silence markers itself)
4. Generate the Corpus from the combined set of training and test data with silence markers (start: <s> and end: </s>) before and after each sentence (This is required by the SLM toolkit)

Errors: The function exits with error condition if an entry in training and/or test data (depending on the selected option) is not found in the phonetic lexicon (Please see section 5.4.6.2.9.1)

5.4.6.2.3 Generate Train IDs and Test IDs

Inputs: Unicode training and test data files, File name prefixes, file number ranges and locations of files on the disk

Outputs: Training and Test IDs files.

5.4.6.2.4 Generate Train and Test Transcription Files

Inputs: Unicode training and test data files, File name prefixes, file number ranges, Phonetic Lexicon

Outputs: Training and Test Transcription files for Sphinx-3

Options: Following options are available for the training and test transcription files

1. Every sentence is transcribed with silence variable (SIL) between all words, and <s> before every sentence and </s>.after every sentence This is useful for non-continuous speech recordings, as we performed in the first step for the phone cover sentences
2. Every sentence is transcribed with <s> before every sentence and </s>.after every sentence This is useful for continuous and spontaneous speech recordings

Errors: The function exits with error condition if an entry in training and/or test data (depending on the selected option) is not found in the phonetic lexicon (Please see section 5.4.6.2.9.1)

5.4.6.2.5 Generate filler dictionary

This is a hard coded function and generates the filler dictionary for the filler words.

5.4.6.2.6 Generate Phone File

This function generates the phone file required by Sphinx-3 and 4, which contains all the phones that are present in the training data.

5.4.6.2.7 Generate Report

This is a very useful function which generates a statistical report about the training and test data. This information is required to predict the baseline for word error rate and also to determine training and testing statistics. Figure 39 shows a sample of the report file. Most of the information is self explanatory however, some parts are described below:

The difference between the entry "No of unique overlapping words between the training and test data:" and the entry "No of overlapping words occurring in the test data:" is that the former gives a count of the number of distinct words in the test data for which the system will be trained by the training data, while the later gives a count of the number of instances of words in the test data for which the system will be trained by the training data.

The four frequency files generated output the extent of training for individual words or phones that the system receives and the extent of testing for individual words that the system will perform. Appendix show samples of all these files for one particular instance of training and testing data.

Inputs: Unicode Training and Test data files, Phonetic Lexicon

Output: Report File and four Word-Frequency and Phone-Frequency files

```
Test Report Generated on: Sat Jun 27 04:01:19 VET 2009

No of Sentences/Utterances in the training file:  1685
No of Sentences/Utterances in the testing file:   50

No of words in the training file:   10677
No of words in the testing file:  313 i.e. 2% of Training Data

No of Unique words in the training file:   1284
No of Unique words in the testing file:  125 i.e. 9% of Training Data

No of Phones in the training file:  36998
No of Phones in the testing file:   1172

No of Unique Phones in the training file: 57
No of Unique Phones in the testing file:   46

No of Unique overlapping words between the Training and test data: 113
No of Unique overlapping Phones between the Training and test data:   46

No of Unique non-overlapping words between the Training and test data:  12
No  of  Unique  non-overlapping  Phones  between  the  Training  and  test  data:
  0

No of Overlapping Words occurring in the test data:  301
No of Overlapping Phones occurring in the test data:   1172

No of Non-Overlapping Words occurring in the test data:   12
No of Non-Overlapping Phones occurring in the test data:  0

Phone to Frequency for training File written to:   Sphinx\Test1\TrP2Fr.txt
Phone to Frequency for testing  File written to:   Sphinx\Test1\TeP2Fr.txt
Word to Frequency for training File written to:    Sphinx\Test1\TrW2Fr.txt
Word to Frequency for testing File written to:  Sphinx\Test1\TeW2Fr.txt
```

Figure 39 - Sample Report File

5.4.6.2.8 Generate Files for Sphinx-4 tests

Two methods provide the required functionality for performing the Wordlist grammar tests on Sphinx-4. The two files “Wordlist.batch” and “WordlistGrammar” are generated using similar methods as employed in producing the dictionary file and the transcription files. The inputs are same as well. The file format is correctly managed by the functions.

5.4.6.2.9 Utility Functions

The following utility functions are developed to assist in the overall corpus generation and Sphinx Files generation process.

5.4.6.2.9.1 Identify new/incorrect words in the Training and Test Data and Suggest Phonetic transcription

This very useful function should be run before running the Sphinx Files compilation. It generates a list of words (if any) that are present in the Training or Test corpus but not in the Phonetic Lexicon. In addition to that it uses the letter to sound rules to generate a suggestion list with the suggested CISAMPA pronunciations for each word. If the words are properly diacritized then the suggestions can be completely correct. Otherwise a manual review can be done to correct any mistakes. These words can then be added to the phonetic lexicon. Sometimes incorrectly typed words or missed spaces etc. are also present in the input corpora, which are also caught at this level.

Inputs: Unicode Training and Test corpora, Phonetic Lexicon, Letter to Sound Rules' Files (three files, for three letter, two letter and one letter sounds)

Output: An exception list file, which lists the new words along with transcription suggestions in CISAMPA. An example of the output is shown in Figure 40.

```
بناسکیں        B N AA S K AEN
پہنچادے        P H N TT_SH AA D_D AE
کردوں          K R D_D OON
قنیچی          Q N II TT_SH II
نجھے           N DD_ZZ_H AE
ٹاون           TT AA OO N
پنتالیس        P N T_D AA L II S
کپمیوٹر        K P M II OO TT R
سافٹیک         S AA F TT II K
گیریژن         G II R II ZZ N
کمپیوٹرسائینس K M P II OO TT R S AA A II N S
کررہاہوں       K R R H AA H OON
فیروزپور       F II R OO Z P OO R
```

Figure 40 - New Words file

5.4.6.2.9.2 Convert from CISAMPA Romanization to Urdu

This functions converts back from CISAMPA based roman script to Urdu Unicode. This can be used to view the recognition results in the Urdu script. It performs a many to one mapping as the multiple pronunciation mechanism in Sphinx in our case is being dealt with in the phonetic lexicon. A word having different pronunciations can be repeated in the lexicon followed by the different Roman and phonetic transcriptions. This reverse conversion function maps all the different Romans back to the same word as mentioned in the phonetic lexicon.

Inputs: CISAMPA based Roman transcription file, Phonetic lexicon

Output: Urdu Unicode file

Errors: The function will give an exception if a word which is not in the phonetic lexicon is encountered.

5.4.6.2.9.3 Generate SOX batch script

Sphinx-4 requires all the audio files in mono channel 16 KHz, 16 bit Big Endean RAW files. Therefore, the test audio files previously in NIST format have to be converted. This function generates a batch script for all the test audio files for the required conversion and sets all the parameters required by Sphinx. The SOX audio conversion utility for Windows® [52] is used by the batch file to perform the required conversion.

5.4.6.2.9.4 Convert Phonetic Lexicon from SAMPA to CISAMPA

This function was developed to convert the Phonetic lexicons which were previously transcribed using standard SAMPA into the new CISAMPA transcription for the purposes of this thesis.

Inputs: Phonetic Lexicon in SAMPA, SAMPA to CISAMPA rules file

Output: Phonetic lexicon transcribed in CISAMPA

Errors: If a non-standard SAMPA symbol is encountered the function gives an error. Solution: New rule(s) can be added to the SAMPA to CISAMPA conversion rules file.

5.4.6.2.9.5 Calculate total Length of NIST files

This function reads the audio file duration from the NIST file header and performs a sum of the durations of the range of files given in the input. It provides an easy way of estimating the actual length of speech files after splitting.

Inputs: Directory path where the files are stored, File name prefix, File name starting number, File Name Ending Number

Output: Displays the total duration of the NIST files in range on the console in minutes and seconds.

5.4.6.2.9.6 Replace all using mapping file

It is often required while cleaning up a new text corpus to batch replace entries in the input corpus e.g. missing spaces etc. as shown in Figure 40 can be corrected once and then replaced all at a time, which save a lot of time. This function helps in the procedure.

Inputs: Tab separated two-column file mapping *words-to-be-replaced* to *new entries,* path of this correction file, path of the old corpus and name of the new corpus.

Output: File with all occurrences of all entries in column 1 of correction file found in the input corpus with the corresponding entries in column 2.

5.4.6.2.9.7 Convert Phonetic Lexicon from CISAMPA based Roman to Grapheme based Roman

The motivation behind this function is explained in 7.5. The task that it performs is to change the old CISAMPA based Romanization scheme in the input phonetic lexicon to the new letter based Romanization as shown in Appendix B on page 139.

Inputs: Path of the old lexicon, path and name of the mapping rules (Appendix B), path and name of new lexicon file to be created

Output: Phonetic Lexicon with only the Romanization changed according to the new rules.

Chapter 6

Experimental Results

6.1 Test Setup

6.1.1 Available Data

The total amount of data that we used for training and testing our system can be summarized as follows:

- 70 minutes of transcribed read speech consisting of 708 greedily made sentences representing all the phones and triphone combinations in Urdu. The data consists of 10101 words, and 5656 unique words. It contains 60 unique phones and 42289 phone occurrences. The word-frequency and phone-frequency graphs of the data are shown in Figure 41. It must be noted that the sentences contained in this corpus are all hand made by trained linguists following the greedy approach to accommodate all the words (which were selected by the set cover algorithm) and to prevent additional words as much as possible. Therefore, while correct grammatically, there are places where these do not represent the semantic structures of Urdu.

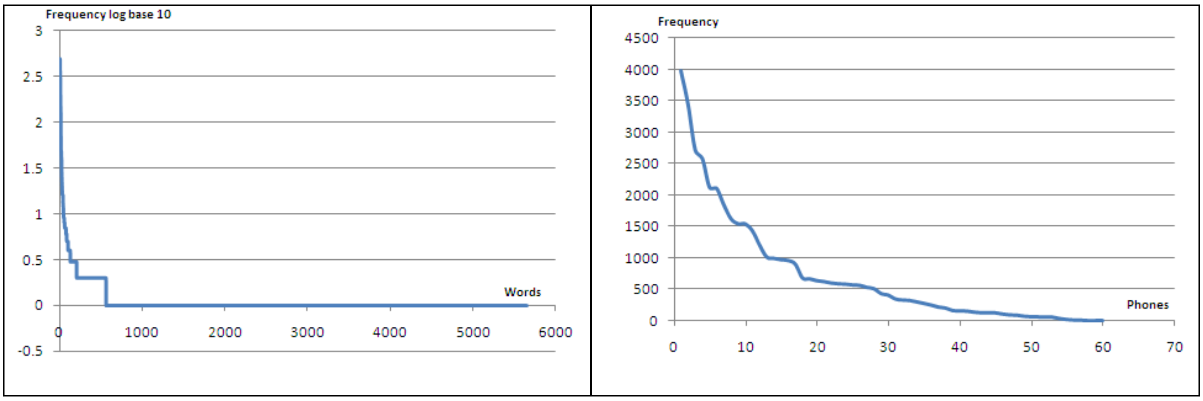


Figure 41 - Word-Frequency (left) and Phone-Frequency (right) graphs of the Read Corpus

- 109 minutes of transcribed spontaneous speech consisting of 3266 utterances recorded in the form of interviews. The data consists of 21034 words and 2032 unique words. It contains 60 different phones with 72700 phone occurrences. The word-frequency and phone-frequency graphs are shown in Figure 42. This data represents the natural and spontaneous speech patterns of a native speaker of Urdu.

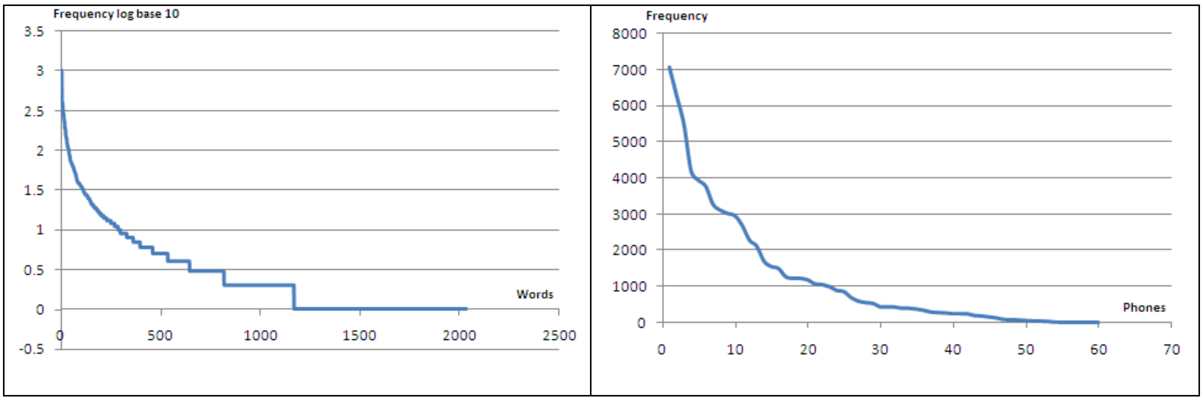


Figure 42 - Word-Frequency (left) and Phone-Frequency (right) graphs of the Spontaneous Corpus

- The combined data from the spontaneous and read (excluding 22 minutes of spontaneous speech data, which is used only for testing purposes) contains 3174 utterances spanning over 157 minutes of speech. It contains 31135 words (6693 unique words), 114990 phones (62 unique phones including the rare L_H). The word-frequency and phone-frequency graphs are shown in Figure 43.

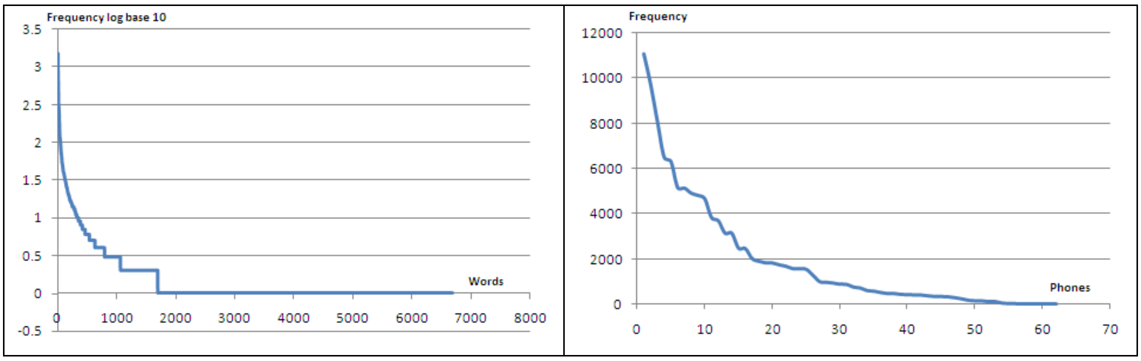


Figure 43 - Word-Frequency (left) and Phone-Frequency (right) graphs of the Combined Spontaneous and Read Corpus

6.1.2 **Test Plan**

The tests were divided into three main portions each designed to achieve a specific goal. These are as follows:

1. **Read Speech**

   Training Data: The system is progressively trained with increasing amounts of read speech in steps of 100 utterances (100, 200… 700 utterances).

   Test Data: For every instance of training, the system is tested with 20% of the size of the training data. These tests are performed in two flavours:

   a. **Baseline Tests:** In this case the system is tested on the training data. The purpose of these tests is to see whether:
      - The system has been properly trained or not and if everything is working as expected
      - To establish the upper bound on system performance as the system should produce the best results if tested on part of the training data
      - To fine tune the ASR decoder properties like Viterbi search beam width, Language Weight and different Smoothing techniques applied to the language model

   b. **Rerecorded Tests:** As mentioned earlier, the read speech cannot be used in making good language models. Therefore, the system cannot be usefully tested with new test data, i.e. data with different grammar and semantic characteristics. Hence, 20% of the written data (for all the seven tests) from within the training data is rerecorded in a different environment. This is then used for testing the system.

   c. **Separate Tests:** These tests were performed with a new set of training data. 100 utterances of everyday read Urdu speech were tested with the system with all 708 utterances of read greedy sentences. The goal was to confirm the hypothesis that the read sentences do provide a good acoustic model but not a good (representative) Language Model for read Urdu speech

2. **Spontaneous Speech**

   These tests were designed to analyse the spontaneous speech recognition characteristics of the system and to fine tune the ASR decoding parameters. The 109 minutes of spontaneous speech data was partitioned into 87 minutes (~80%) of training data and 22 minutes (~20%) of test data. The language model is derived from the training data only and the system is tested with different parameters of the decoder.

3. **Mixture of Read and Spontaneous Speech**

   These experiments were devised with the goal of finding the optimal spontaneous to read data ratio that would give best recognition results on spontaneous speech. The system is now tested with 800 utterances (22 minutes) of spontaneous (completely non-overlapping with any of the training data). The system is then progressively trained with 100% of spontaneous speech + *x* % of read speech (where *x* increases in steps of 25 % from 0 to 100%). Next the system is trained with a mixture of 100% read speech + *x*% of spontaneous speech (where *x* increases in steps of 25% from 0 to 100%). All other parameters are kept constant to observe the required trend only.

Note: *The value of beam width is represented in Sphinx as a number between 0 and 1, with 0 as the widest (no paths are pruned) and 1 as the narrowest (only the best path survives pruning). Therefore the values of beam width are written as 1e-x (i.e. 1x10^{-x}). Therefore, as x increases in magnitude, the beam width increases.*

6.1.3 Test Set – 1 (a & b): Read Speech

6.1.3.1 100 Sentences (10 minutes)

In the first test 100 utterances (~10 minutes of speech) from the read corpus were used to train the system. The system was then tested with two different sets of test data:

- **Control Set:** 20 sentences from the training data itself
- **Rerecorded Set:** The same 20 sentences rerecorded in an environment different from the training environment

Detailed statistics of training and test data are given in Table 4.

	Training Data	Test Data
No. of utterances	100	20
Duration (minutes)	10	2
No. of words	1374	341
No. of unique words	875	244
No. of Phones	5733	1429
No. of unique Phones	55	53

Table 4 - Training and Test data (100 utterances)

The Trigram language model was developed from the 100 utterances of the training data using the Online Language Modelling Toolkit (OLMT), and absolute discounting was used as the smoothing technique. Initially the language weight was fixed at 6 and the beam width was modified from 1e-100 through 1e-700 with the results shown in Table 5 and Figure 44.

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-100	6	36.16	65.86	53.94	65.86
1e-200	6	23.84	78.99	51.11	69.09
1e-300	6	24.24	79.39	51.52	69.09
1e-400	6	15.15	89.70	51.72	68.48
1e-500	6	15.15	89.70	51.72	68.48
1e-600	6	15.15	89.70	51.72	68.48
1e-700	6	15.15	89.70	51.72	68.48

Table 5 – Effects of Beam Width Variation (100 utterances)

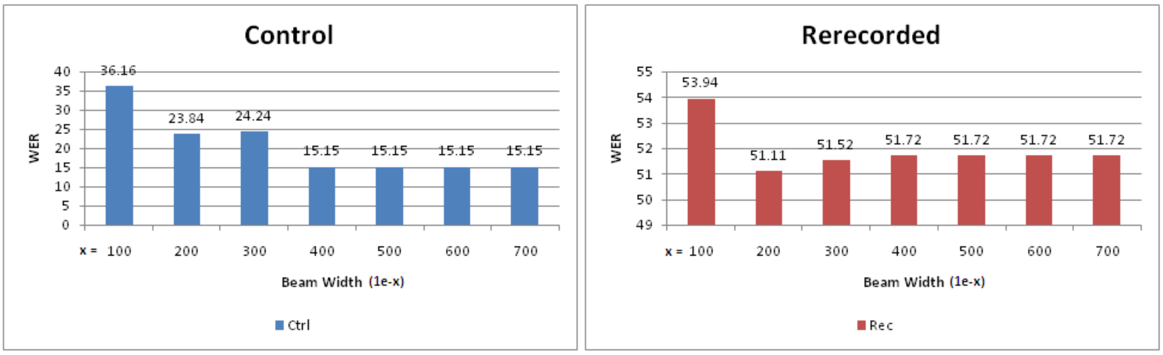


Figure 44 – Effects of Beam Width Variation (100 utterances)

The tables show that the best WER of 15.15% is achieved for beam widths greater than 1e-400 for the control experiments and a WER of 51.11% for rerecorded speech using a beam width of

1e-200. Next the experiments were repeated with a fixed beam width (BW) of 1e-700 and language weight was modified. The recommendations of Sphinx tutorials are to use language weights between 6 and 13. These values along with two outlying values were tried. The results have been shown in the Table 6 and Figure 45.

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-700	1	14.75	90.71	66.46	62.63
1e-700	7	13.54	90.10	50.71	67.88
1e-700	8	13.54	90.10	51.92	66.67
1e-700	9	13.33	90.10	50.91	66.26
1e-700	10	12.93	90.10	50.51	66.46
1e-700	11	13.13	89.90	49.70	65.45
1e-700	12	13.13	89.70	48.69	65.25
1e-700	24	14.95	87.07	53.94	49.49

Table 6 – Effects of Language Weight Variation (100 utterances)

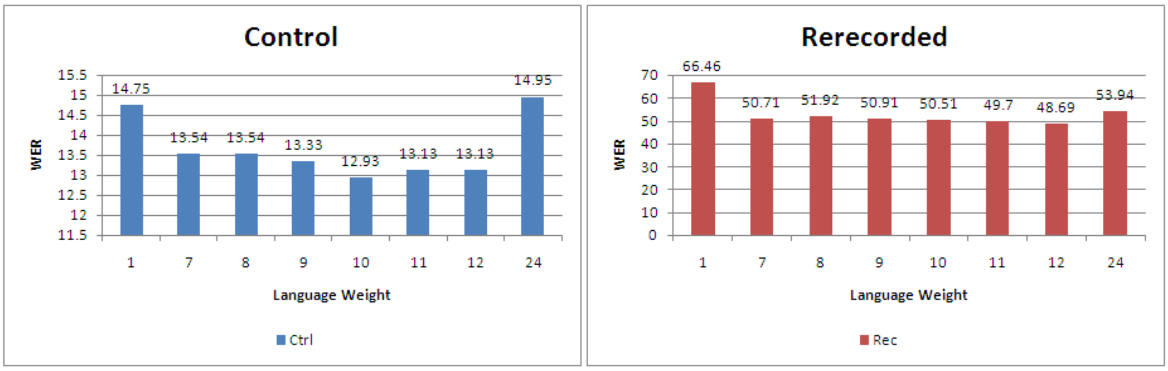


Figure 45 – Effects of Language Weight Variation (100 utterances)

The language weight (LW) value of 10 gives the best WER value of 12.93% for the control tests. While the language weight of 12 performs the best for rerecorded speech by giving a WER of 48.69%. However, the overall outcome of these tests indicates (from the high values of WERs) that the training data is not yet enough to train the system to recognize the 53 phones present in the test data. As a result, the validity of the beam width and language weight parameters as being the most optimal ones remains uncertain. Therefore, the next logical step was to increase the size of the training data (and also the test data proportionally) and to run through all these tests again to find out the best values of Beam Width and Language Weight.

6.1.3.2 200 Sentences (20 minutes)

The experiment was repeated with 200 utterances of read speech as training data. The statistics to training and test data are mentioned in Table 7. The trigram language model has been developed from the training data using the online Language Modelling Toolkit [51] with absolute discounting. Like before the test data consists of two parts:

- **Control Set:** 40 sentences from the training data itself
- **Rerecorded Set:** The same 40 sentences rerecorded in an environment different from the training environment

	Training Data	Test Data
No. of utterances	200	40
Duration (minutes)	20	4
No. of words	2764	618
No. of unique words	1698	418
No. of Phones	11870	2718
No. of unique Phones	58	54

Table 7 - Training and Test Data (200 utterances)

The initial tests were repeated as in the previous part to focus on the optimal beam width while keeping the language weight constant as shown in Table 8 and Figure 46.

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-100	6	0.66	99.34	14.66	90.30
1e-200	6	0.66	99.34	14.88	90.41
1e-300	6	0.66	99.34	15.55	89.31
1e-400	6	0.66	99.34	15.55	89.75
1e-500	6	0.66	99.34	15.55	89.75
1e-600	6	0.66	99.34	15.55	89.75
1e-700	6	0.66	99.34	15.55	89.75

Table 8 - Effects of Beam Width variation (200 utterances)

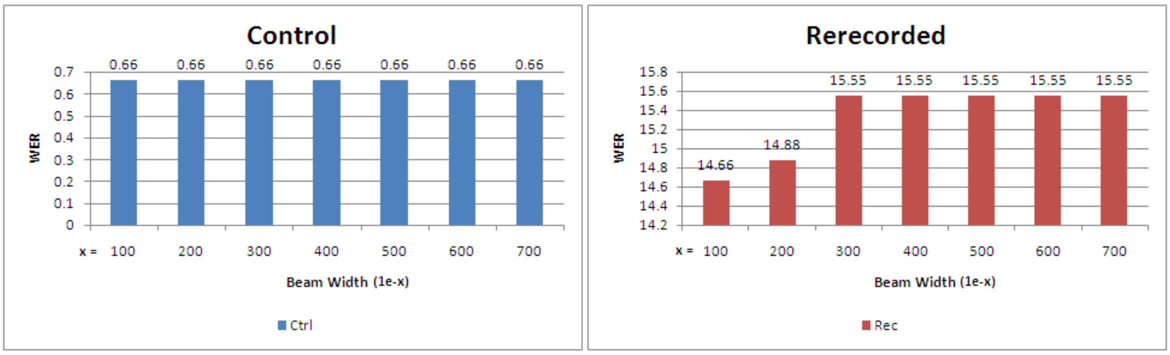


Figure 46 - Effects of Beam Width Variation (200 utterances)

The results show a drastic improvement for 200 utterances. Now the WER for the control data seems more beam width independent as its value remain the same for all values of the test beam widths. This is understandable as the control test data was an overlapping part of the training data, therefore the correct paths should ascend to higher probabilities in the Viterbi. As a result even smaller beam width would perform equally well as larger ones. For the rerecorded speech the matters are a bit different. The results are slightly better for smaller Beam Widths while worse for larger. This problem has been indicated in various texts as a negative aspect of too large beam widths which prevent less probable paths from being pruned at earlier stages. These paths may gain probability as the algorithm proceeds due to some incorrect utterances later in the sentence giving rise to local maxima of path probabilities, hence making the WER slightly greater in certain cases. This reasoning is supported by the fact that the WER reaches a plateau after Beam Widths of 1e-300.

Next the beam width was kept constant at 1e-700 and the experiments were repeated with different values of Language Weight as shown in Table 9 and Figure 47.

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-700	1	0.66	99.67	26.79	85.45
1e-700	7	0.66	99.34	14.22	90.63
1e-700	8	0.88	99.12	13.56	90.96
1e-700	9	0.88	99.12	13.56	90.85
1e-700	10	0.88	99.12	14.00	90.30
1e-700	11	0.77	99.23	13.67	89.86
1e-700	12	0.77	99.23	13.78	89.75
1e-700	24	0.44	99.56	19.96	81.81

Table 9 - Effects of Language Weight (200 utterances)

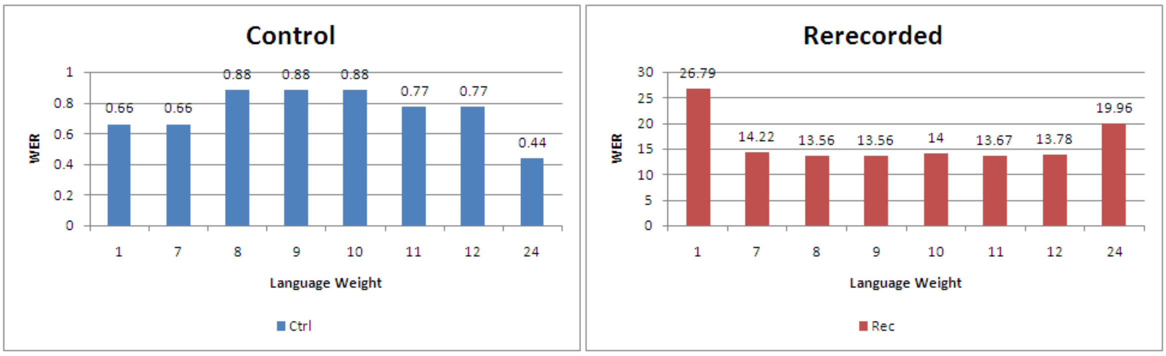


Figure 47 - Effects of Language Weight (200 utterances)

The best values of language weight are achieved for smaller values of language weight and very large values. As the test data has been taken from the training data itself therefore, this is the result of the fact that the acoustic model or language model alone will give higher values of path probabilities then the product of these if likelihood and prior are given equivalent weight. Therefore, the WER is less when prior is given more weight (LW=1 through 7) and when only likelihood is give high weight (LW = 24). The results for rerecorded speech are as expected, with the best WER of 13.56% for LW of 8 and 9.

From these initial experiments it was concluded that there are no great fluctuations in the WER as the BW and LW are modified, therefore, the resolution of these factors is made coarser in the next experiments. The goal of these experiments is to find the optimal values of beam width and language weight.

6.1.3.3 300 Sentences (30 minutes)

The system was trained with 300 utterances comprising a total of 30 minutes of speech. The language model was developed using the online language modelling toolkit with absolute discounting. The details of training and test data are mentioned in Table 10.

	Training Data	**Test Data**
No. of utterances	300	60
Duration (minutes)	30	6
No. of words	4359	937
No. of unique words	2552	617
No. of Phones	18266	4051
No. of unique Phones	58	55

Table 10 Training and Test Data (300 utterances)

The system is then tested with three different values of beam width, while keeping the language weight fixed at 7. Next the beam width is made constant and the experiments are repeated with three different values of language weight. The results on control and rerecorded data are shown in the Table 11.

Beam Width	**Language Weight**	**Control**	**Control**	**Rerecorded**	**Rerecorded**
		WER %	**% Correct**	**WER %**	**% Correct**
1e-100	7	0.95	99.42	11.55	93.13
1e-300	7	1.02	99.42	11.55	93.42
1e-500	7	1.02	99.42	11.84	93.13
1e-700	7	1.02	99.42	11.84	93.13
1e-700	9	0.88	99.56	11.84	92.84
1e-700	11	1.32	99.20	11.62	92.62

Table 11 - Test Results (300 utterances)

The results seem to be pretty stable overall. There are only minor fluctuations in the WERs. For control data the best values are obtained for BW = 1e-700 and LW = 9. For rerecorded, the best values seem more dependent on language weight than beam width and good results are obtained for LW = 7 and LW = 11.

6.1.3.4 400 Sentences (40 minutes)

The same series of tests that were performed for 300 sentences is repeated for 400 sentences with one major difference. The Trigram LM is now developed using the SLM Toolkit, as the online LM toolkit does not support more than 5000 tokens. The discounting method used is Witten-Bell discounting. The statistics of test and training data are shown in Table 12 and the test results are mentioned in Table 13.

	Training Data	Test Data
No. of utterances	400	80
Duration (minutes)	40	8
No. of words	5900	1276
No. of unique words	3372	823
No. of Phones	24514	5377
No. of unique Phones	58	57

Table 12 - Training and Test Data (400 utterances)

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-100	7	1.95	98.59	7.70	95.34
1e-300	7	1.52	99.02	7.70	95.34
1e-500	7	1.52	99.02	7.92	95.12
1e-700	7	1.52	99.02	7.92	95.12
1e-700	9	1.57	98.97	7.59	95.17
1e-700	11	1.74	98.81	7.21	94.9

Table 13 - Test Results (400 utterances)

The test results for rerecorded speech now show a marked improvement especially for language weights of 9 and 11 and a beam width of 1e-700. On the other hand the results for control data are best for smaller beam widths and smaller values of language weight.

However, the statistics obtained from these experiments clearly indicate that for the rerecorded speech better values of WER are being obtained for beam widths in the ranges of 7 to 9 and for language weights of 1e-500 to 1e-700. Since the system performs much faster with a beam width of 1e-500 as compared to 1e-700, therefore the remaining experiments are done with a beam width of 1e-700 and a fixed language weight of 8.

6.1.3.5 500 Sentences (50 minutes) and 600 Sentences (60 minutes)

Table 14 and Table 16 show the statistics of the training and testing data for the next two experiments with 500 and 600 sentences respectively. In both the tests Trigram LMs developed using the SLM Toolkit with Witten-Bell discounting are employed. Table 15 and Table 17 show the test results.

	Training Data	Test Data
No. of utterances	500	100
Duration (minutes)	50	10
No. of Words	7294	1555
No. of Unique Words	4129	997
No. of Phones	30706	6597
No. of Unique Phones	58	57

Table 14 - Training and Test Data (500 utterances)

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-500	8	2.62	97.86	7.21	95.15

Table 15 - Test Results (500 utterances)

	Training Data	Test Data
No. of Utterances	600	120
Duration (minutes)	60	12
No. of Words	8588	1801
No. of Unique Words	4847	1147
No. of Phones	36378	7676
No. of Unique Phones	59	58

Table 16 - Training and Test Data (600 utterances)

Beam Width	Language Weight	Control	Control	Rerecorded	Rerecorded
		WER %	% Correct	WER %	% Correct
1e-500	8	3.1	97.7	8.65	93.53

Table 17 - Test Results (600 utterances)

The results show a increase in the WER for control experiments and also a slight increase in WER for rerecorded speech. It shows that as the training model is becoming more general there is a trend towards fall in performance as far as WER is concerned. Another reason can be the choice of lower beam width of 1e-500 and 1e-700 may have given slightly better results showing a plateau of performance.

6.1.3.6 708 Sentences (70 minutes)

The final test with overlapping read speech is done using all the 708 utterances for the training data and the same parameters for BW and LW as were used in the previous two experiments. The details of Training and Test data are shown in Table 18

	Training Data	Test Data
No. of Utterances	708	140
Duration (minutes)	70	14
No. of Words	10101	2106
No. of Unique Words	5656	1317
No. of Phones	42289	8906
No. of Unique Phones	60	58

Table 18 - Training and Test Data (708 utterances)

The system is now tested with all the discounting strategies provided by the SLM Toolkit. Trigram language models are used. The results are shown in the Table 19

Beam Width	Language Weight	LM Smoothing	Control	Control	Rerecorde d	Rerecorde d
			WER %	% Correct	WER %	% Correct
1e-500	8	Absolute	4.56	96.16	10.41	91.44
1e-500	8	Good-Turing	4.85	95.87	11.1	90.82
1e-500	8	Linear	4.36	96.32	8.46	93.17
1e-500	8	Witten Bell	3.68	96.81	8.14	93.3

Table 19 - Test Results (708 utterances)

The results indicate that for rerecorded speech the best WERs are achieved with Witten-Bell discounting strategy closely followed by the linear discounting. This precise behaviour is shown by the control data as well.

This completes the tests with the sentences obtained by greedy strategy from words which provide phonetic cover and phonetic balance. The next section summarizes the results.

6.1.3.7 Summary of Tests

Table 20and Figure 48 show the summary of the best results obtained as the training data was increased from 100 to 708 utterances while testing with different values of beam width and language weight. The optimal value of beam width is found to be 1e-500. This value is better as it provided a nice balance between efficiency and performance with reference to WER. The optimal language weight has been found to be 8 for these experiments.

Combined Results	Control	Rerecorded
Training data Size (No. of sentences)	**Best WER %**	**Best WER %**
100	12.93	48.69
200	0.44	13.56
300	0.88	11.55
400	1.52	7.21
500	2.62	7.21
600	3.10	8.65
700	3.68	8.14

Table 20 - Combined Read Speech Test Results

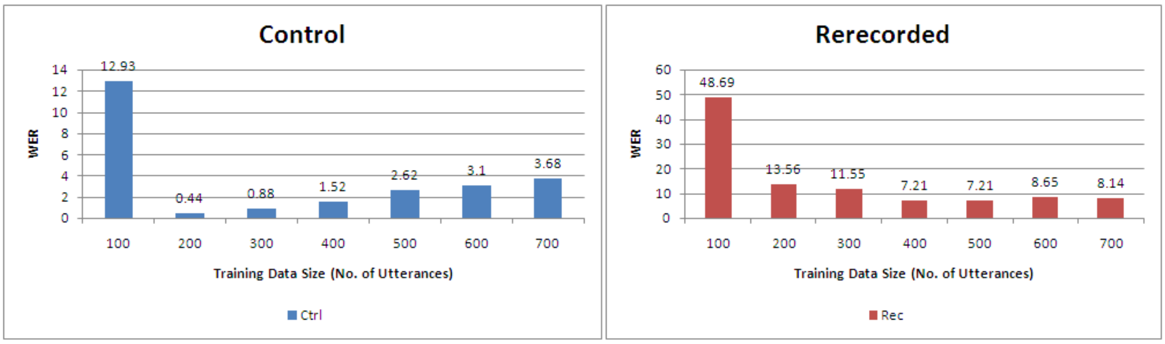


Figure 48 - Combined Read Speech Test Results

An analysis of the test summary shows that after the initial bad performance for 100 utterances of training data, the WER comes within acceptable limits and stays there for all the rest of the values. The control experiments show that the system performs at its peak value within 4% of WER at maximum for the most general training set. The rerecorded test data show the adaptation pf the system to different test environmental conditions as the speaker and test corpus remains the same. Moreover, the trend in the rerecorded test data is continuously towards improvement as training data is increased, with a slight fluctuation in WER at 600 sentences.

Next the system is tested with a new set of read data to see its response to new speech.

6.1.4 Test Set – 2: Part c Read Speech

In this experiment the system trained with the 708 sentences of read speech is tested with 100 utterances (nearly 10 minutes) of test data that is completely separate from the training data set. The sentences are from newspapers and other sources of print media. The statistics of training and test data are shown in Table 21.

	Training Data	Test Data
No. of utterances	708	100
Duration (minutes)	70	10
No. of words	10101	752
No. of unique words	5656	444
No. of Phones	42289	2885
No. of unique Phones	60	55
Out of Vocabulary Words	-	185
Instances of OOVs	-	211

Table 21 - Training and Test Data (Non-Overlapping tests)

The out of vocabulary (OOV) words in the test data were added to the phonetic dictionary of the training data. However, the language model is generated from the training data alone so it does not contain those words. As a result 28% of the test data is not represented in the language model. The WER achieved was 53.50%. The main reason for the poor WER is the presence of a large number of OOVs and also that the LM generated from the read training corpus are not representative of the natural grammatical and syntactic structures of Urdu speech. The 46.5% recognition can be attributed to the well trained acoustic model. The language weight used in the experiment was 8 and the beam width was fixed at 1e-500.

This experiment clearly shows the need of a language model which should represent the Urdu speech and its structures more precisely and should be large enough to model good N-gram probability estimates.

6.1.5 **Test Set – 3: Spontaneous Speech**

This third set of experiments is designed to analyze the performance of the ASR when it is trained with spontaneous speech only and is tested with spontaneous data. The training and test data are in 80:20 ratios and the trigram language model is obtained only from the training spontaneous data corpus. As a result there are a lot of out of vocabulary words. The spontaneous speech corpus is far from being sufficient for providing a representative language model yet, it is the best that is available for now. The statistics of training and test data are shown in Table 22 and it can be clearly seen that the OOV instances comprise 10% of the test data. The good thing is that the training data provides a good phonetic cover and is phonetically balanced as well as was shown in Figure 42 (right). This is due to the simple reason that all this speech has been spontaneously spoken by a native speaker. The duration of training data is a little short of one and a half hour and should provide good enough training as a start.

	Training Data	**Test Data**
No. of utterances	2466	800
Duration (minutes)	87	22
No. of words	21034	4623
No. of unique words	2032	750
No. of Phones	72700	16442
No. of unique Phones	60	55
Out of Vocabulary Words	-	212
Instances of OOVs	-	471

Table 22 - Training and Test Data (Spontaneous Speech)

All the tests results shown in Table 23 are results of tests with a fixed beam width of 1e-700. This was done to ensure a good enough Viterbi search space. The tests have been conducted to finalize the appropriate language weight. The language model was smoothed using Witten-Bell discounting that has shown the best results for the tests so far. It took around 1.5 hours to train the system for the 87 minutes of speech and a further 45 minutes (approximately) for running each test. The tests were conducted on a Toshiba Satellite (model number: m115) Laptop computer with a 1.6GHz Dual Core processor and 2.5 Giga bytes of RAM.

Language Weight	WER
6	24.9
8	22.9
9	23.5
11	22.8
13	23

Table 23 - Spontaneous Speech Test Results with fixed Beam Width of 1e-700

The results indicate that better WER values are obtained for language weight values of 8 and 11. Overall the results lie within ranges of 22 and 25, which is not bad considering the large number of OOVs and a language model derived from a small corpus.

6.1.6 Test Set – 4: Mixture of Read and Spontaneous Speech

The final run of tests was conducted to find the optimal spontaneous to read speech training data ratio that would give the best results for recognizing spontaneous speech. The intuition behind these experiments is that the greedily generated read speech should provide enough phonetic cover and balance for a good acoustic model. In addition further training with spontaneous speech should not only provide better modelling for acoustic model but also a good overall language model representing the spontaneous speech of Urdu.

The experiments involve 87 minutes of spontaneous speech training data and 70 minutes of read speech training data. The test data consists of 22 minutes of spontaneous speech entirely separate (non-overlapping) from the training data. The statistics of the two types of training data and the test data are mentioned in Table 24.

	Spontaneous Training Data	Read Training Data	Spontaneous Test Data
No. of utterances	2466	708	800
Duration (minutes)	87	70	22
No. of words	21034	10101	4623
No. of unique words	2032	5656	750
No. of Phones	72700	42289	16442
No. of unique Phones	60	60	55

Table 24 - Training and Test Data for Sp:Re Ratio Experiments

The experiments were performed using two different types of language models. One (which will be referred to as spontaneous LM below) is derived from the 87 minutes of transcribed spontaneous speech only. The recognition results obtained with this language model are shown in Table 25 along with the details regarding number of unique out of vocabulary words (OOVs), the number of instances of OOVs in the test data and the number of occurrences of words in the test data which are not present in the language model. The language model in all cases is a trigram language model with Witten-Bell discounting generated using the SLM Toolkit. The tests are performed on systems trained with the two types of training data mixed together in different ratios. The recognition results are summarized in Figure 49. Figure 50 and Figure 51 depict the relationship between WER and OOVs and WER and OOV instances for different training ratios. In all tests the beam width used is 1e-700 and language weight of 8. Hence, all other factors except the spontaneous to read ratio are maintained constant.

Training Data (Spontaneous:Read)	WER Spontaneous LM	OOVs	OOV Instances	LM OOVs
100:0	22.9	212	471	212
100:25	23.1	182	410	212
100:50	23.4	168	347	212
100:75	23.8	154	324	212
100:100	23	136	279	212
75:100	23.9	151	329	212
50:100	23.9	174	384	212
25:100	26.8	209	445	212
0:100	75.3	297	826	212

Table 25 - Results with Spontaneous Language Model

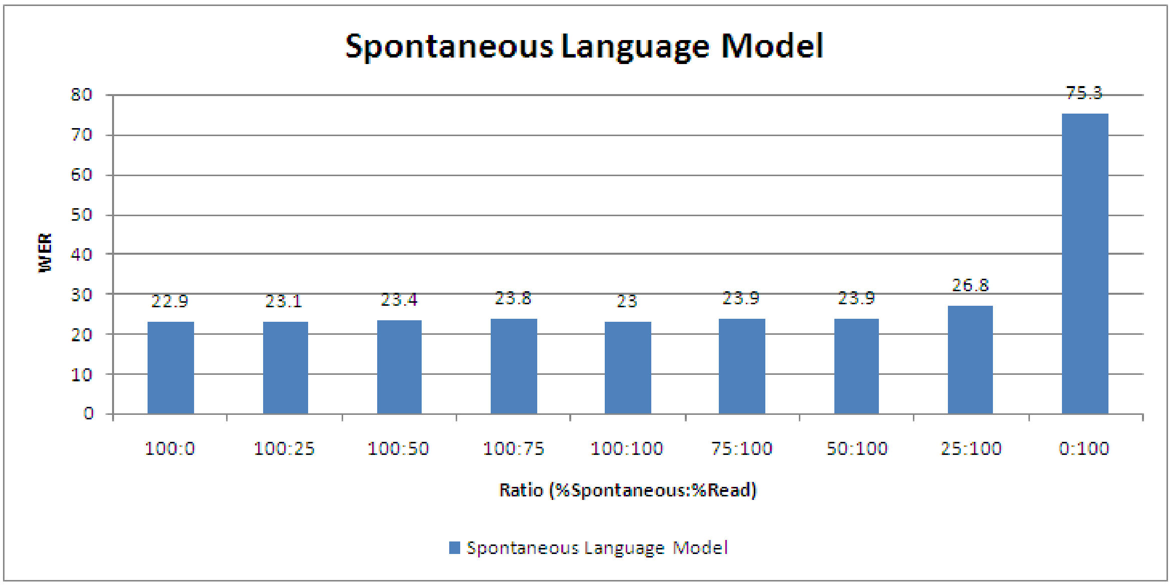


Figure 49- Results with Spontaneous Language Model

Figure 49 shows that using the spontaneous speech language model the error increases slightly as read speech data is included into the spontaneous speech data. However the WER reaches a satisfactory number 23% for a 1:1 ratio between spontaneous and read data. After that however, it starts ascending much rapidly then it fell to the valley. As the spontaneous data portion becomes less in the mixture the WER increase ultimately ending on a high 75.3% for read speech data alone. Therefore it can be concluded that including read speech to spontaneous speech while the language model is being derived from spontaneous speech alone does not improve the recognition results. However, this can be a result of the non-overlapping words between the read and spontaneous corpora which means that those words will not be present in the language model either and hence will have a very low probability towards recognition.

A second point to be noted is that while increasing training data (as we moved from 100:0 towards 100:100) the recognition results constantly became better (except for the initial 22.9%). This may mean that the system is still in need of more training data for better training. The apparent valley of WER obtained at 100:100 may become better if the amount of speech data is increased. This, however, can only be proven by further training and testing iterations.

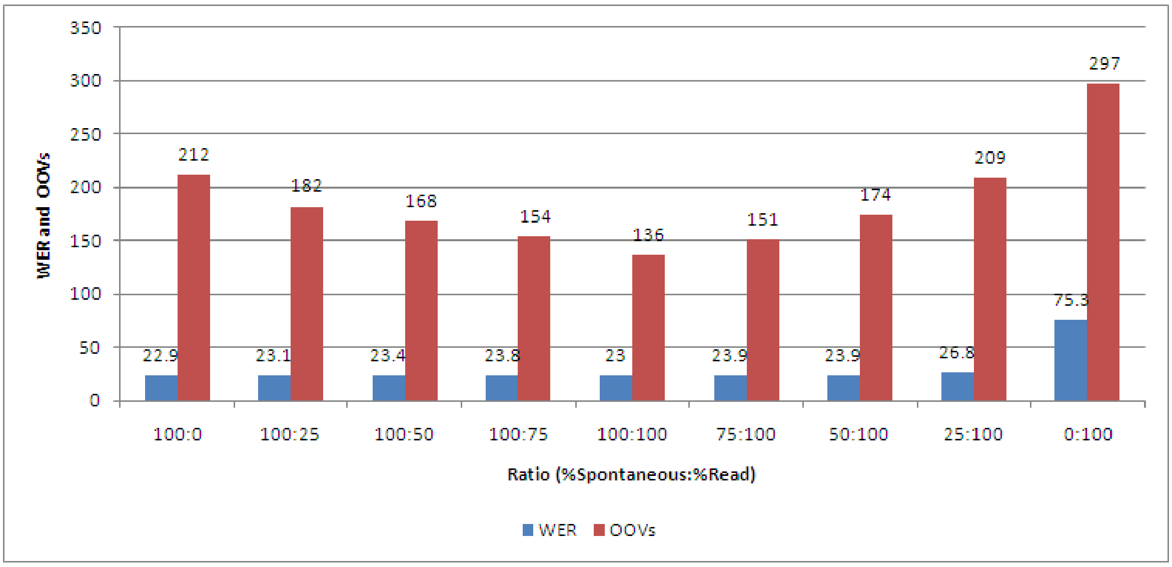


Figure 50 - Comparison of WER with OOVs in the Test Data

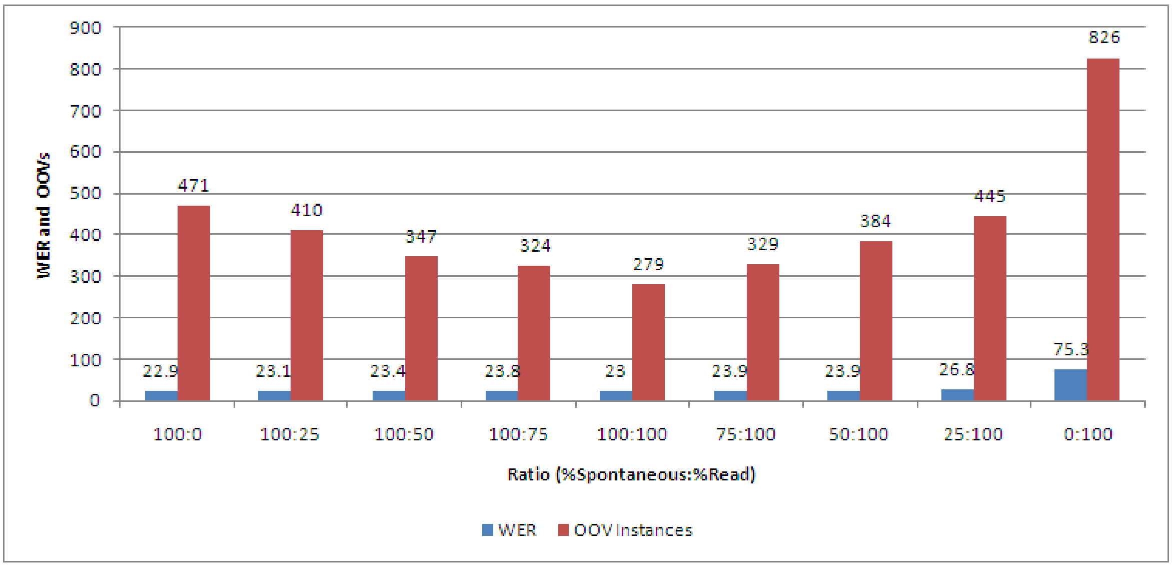


Figure 51 - Comparison of WER with OOV instances in the Test Data

Figure 50 and Figure 51 show the relation ship between WER and OOVs. It is clear that with the exception of the initial 22.9% WER for spontaneous data alone, the WER directly proportional to the OOVs. The minimum WER of 23% aligns with the minimum OOVs of 136 and minimum

OOV instances of 279. The cross correlation between WERs and OOVs gives a 0.86 and between WER and OOV instances gives 0.93. Hence, it reaffirms the strong dependency of OOVs on WER. The sharp peak of 75.3% WER at 0:100 ratios between spontaneous and read data correspond with a sharp rise in OOVs and hence explain the reason for the sudden increase in WER. However, it must be mentioned that this OOV amount just reflects on the acoustic model and not the language model which is a constant factor during the tests with a fixed OOV count of 212 and OOV instance count of 471.

The second set of experiments was performed using language models derived from the actual training data (which will be referred to as training LM below). Therefore, the LM varies from test to test as the ratio between the spontaneous and read speech varies in the training data. The recognition results obtained with this language model are shown in Table 26 along with the details regarding number of unique out of vocabulary words (OOVs), the number of instances of OOVs in the test data and the number of occurrences of words in the test data which are not present in the language model.

Training Data (Spontaneous:Read)	WER Training LM	OOVs	OOV Instances	LM OOVs
100:0	22.9	212	471	212
100:25	21.5	182	410	182
100:50	21.0	168	347	168
100:75	20.3	154	324	154
100:100	18.8	136	279	136
75:100	22.1	151	329	151
50:100	23.7	174	384	174
25:100	29.1	209	445	209
0:100	58.4	297	826	297

Table 26 - Results with Training Data based Language Model

It can be observed that the LM OOVs change from test to test as the training data changes. The language model in all cases is a trigram language model with Witten-Bell discounting generated using the SLM Toolkit. The tests are performed on systems trained with the two types of training data mixed together in different ratios. The recognition results are summarized in Figure 52. Figure 53 and Figure 54 depict the relationship between WER and OOVs and WER and OOV instances for different training ratios. In all tests the beam width used is 1e-700 and language weight of 8. Hence, all other factors except the spontaneous to read ratio are maintained constant.

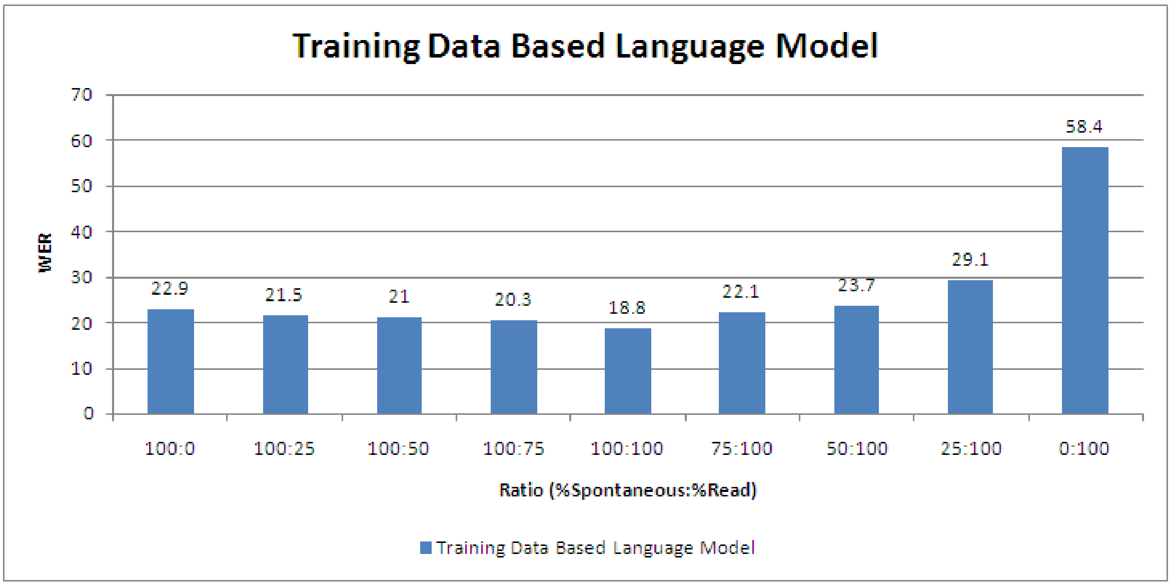


Figure 52 - Results with Training Data based Language Model

The results for the training data based language model show a very nice trend and very clearly depict the effects of the ratio on recognition results. It can be seen that the WER starts decreasing as the read speech is introduced into the mixture of training data hence increasing the over all amount of data as well. The WER reaches a minimum of 18.8% for the 1:1 ratio between spontaneous and read speech and then begins to climb rapidly as the spontaneous data becomes limited in the mixture. Finally reaching a high WER of 58.4 for read speech based training data. The results are very similar to the ones obtained for the spontaneous speech LM only better. However, the hypothesis that the system is still in need of more training data in term of duration and amount is reinforced as we can see that we obtained the least WER for the maximum amount of over all training data.

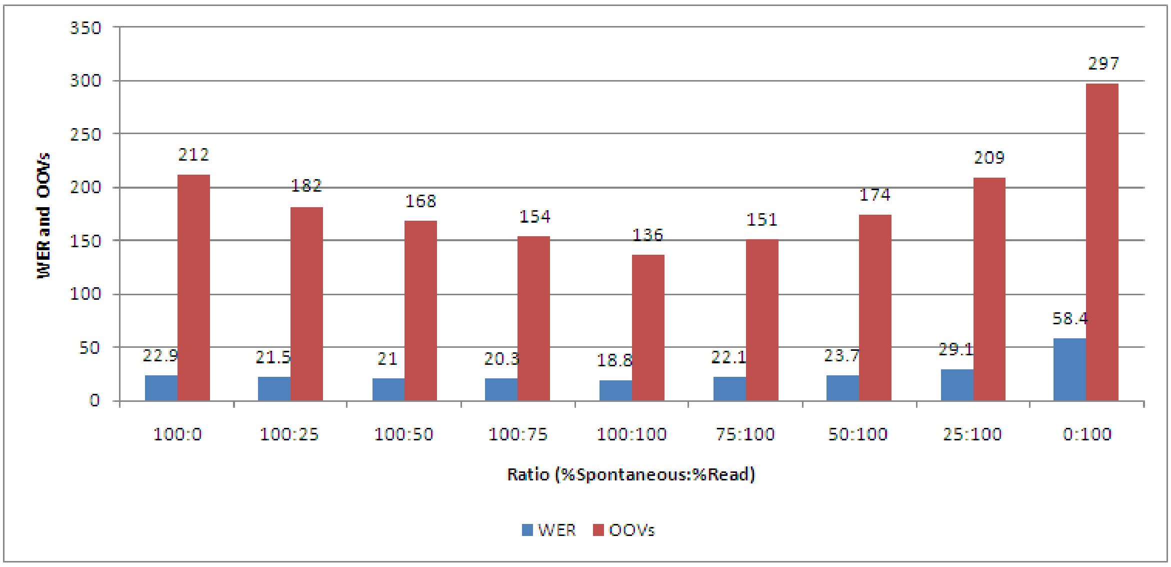


Figure 53 - Comparison of WER with OOVs in the Test Data

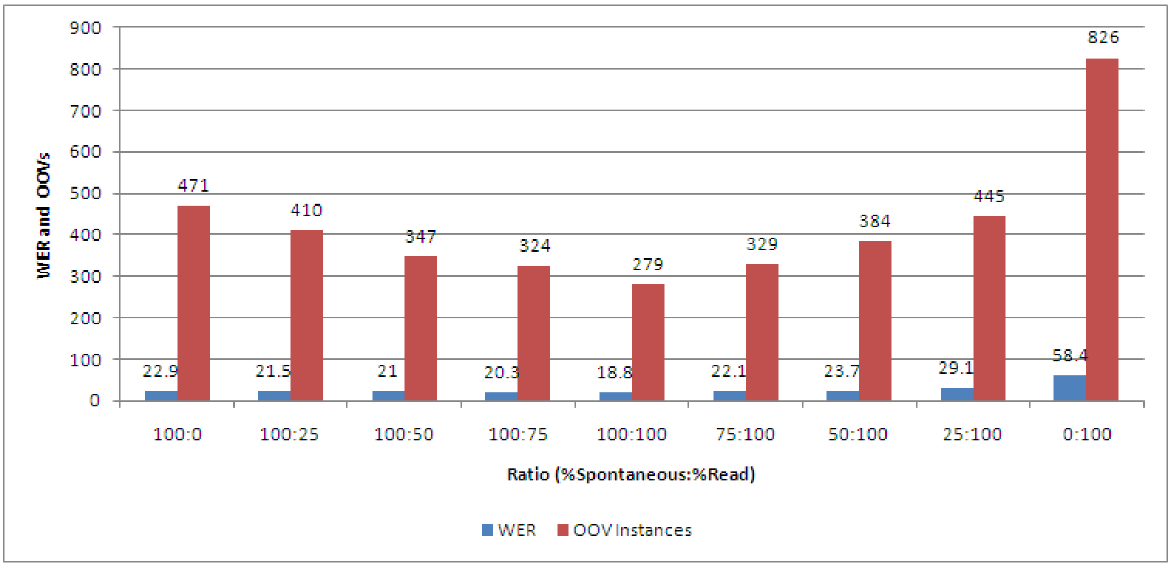


Figure 54 - Comparison of WER with OOV instances in the Test Data

Figure 53 and Figure 54 show the relationship between OOVs and OOV instances and WER. The direct proportionality relation remains valid here as well. The correlation between WERs and OOVs gives a very high 0.92 and that between WERs and OOV instances gives 0.96. It is also notable that now the OOVs also mean that the language model does not contain those words either. The sharp peak of WER 58.4%, again corresponds with the highest OOV and OOV instance value. The WER is relatively less compared to the spontaneous LM tests. The explanation for this behaviour is the large non-overlap between the vocabulary of training data and the language model that existed in case of spontaneous LM when it was tested with 0:100 ratios between training and test data. No such non-overlap exists here as the language model is being derived from the training data and hence the results remain overall better. This fact points towards the importance of matching between training data and the language model. The high correlation between word error rates and out of vocabulary words indicates that a further decrease in WER may be achievable by decreasing the OOVs, which can be accomplished by expanding the corpus for training data and language model.

6.2 Comparison with previous work

With all the different types of speech recognition systems available and under development for different languages it is not an easy task to establish an exact comparison. There are speech recognition systems designed to recognize different types of speech ranging from simple isolated-digits based recognition to spontaneous speech recognition with disfluencies. The environmental noise varies from microphone based quite studio recordings to busy street noise based cell phone speech recognition systems. Then there is great variation regarding the amount of training data which ranges from few minutes of training data to several hundred hours of speech data used for training the ASR systems. Then another factor introducing a lot of variation is the amount of research done on a particular problem regarding some specific language and the stage of development at whish a system currently is.

With all these variations an exact comparison with our speaker specific, spontaneous speech recognition system for Urdu is not possible. However, to get things into perspective and have a rough idea of the performance of our system a comparison is shown with some of the most closely matching system, which were explained in detail in the literature review section. It can be seen that in this comparison with respect to the WER this system shows performance next only to the transcription system for Arabic Broadcast News which cannot be categorized as purely spontaneous. Secondly, all other systems mentioned in the table are the results of improvements by using different techniques. The 18.8% WER shown by our system is only the simple and unimproved performance, which may be improved further simply by adding more training data. Aside from this table we know that the WERs for the best available spontaneous speech recognition systems are around 15% for broad cast news [35] and [36] and 40% for meeting and telephone conversation transcription [37].

On the other hand, all other systems shown in the comparison are speaker independent, which is one aspect that our system has not been trained and tested for. In short, it can be said that this system has shown a promising performance for its initial run and may improve further by applying simple techniques (some of which are suggested in the Future Directions).

Paper	**ASR Engine**	**Type**	**Best WER %**
Soltau et al [37]	GALE ASR	Transcription of Arabic broadcast news	14.9
Our System	**Sphinx-3**	**Speaker specific Speech recognition system for spontaneous Urdu Speech**	**18.8**
Digalakis et al [22]	SRI's DECIPHER	Greek Dictation System	19.27
Raškinis et al [27]	HTK	Speech recognition system for Lithuanian on an isolated word phonetically rich corpus	20
Gauvain et al [32]	LIMSI	Speaker independent continuous speech recognition system for transcribing unrestricted American English broadcast news	20
Gauvain et al [32]	LIMSI	Speaker independent continuous speech recognition system for transcribing unrestricted American English broadcast news	20
Anumanchipalli et al [25]	Sphinx-2	Tamil, Telugu and Marathi landline and cellular phone based continuous (read) speech recognition system	23.6
Takaaki et al [34]	-	A method for paraphrasing spontaneous Japanese speech into written style sentences	24.2
Frank et al [36]	-	Transcription of continuous broadcast news data	29.3
Jacques et al [30]	ESAT	Spontaneous English Telephone Calls	29.6

Vivek et al [35]	-	Disfluent repetitions in spontaneous speech	42.1
Nedel et al [38]	Sphinx-3	ASR for spontaneous English speech	49.3

Table 27 - Comparison of ASR systems

6.3 **Conclusion**

The aim of this thesis was to develop a speaker specific speech recognition system for spontaneous and read Urdu speech. The main hurdle was the lack of speech corpus and transcribed speech resources. Therefore, the first goal of this project was to develop a phonetically rich and balanced sentence based text corpus for Urdu providing context based phonetic (with triphoneme as the phonetic context unit) cover for Urdu speech. The corpus was developed using greedy approach with acoustic phonetic enhancements to reduce its size and to make the resulting data set more natural. This corpus was read and recorded to produce the read speech corpus for Urdu. Next a set of interviews was designed and recorded to produce the spontaneous speech corpus which was manually transcribed. Using the Sphinx-3 Automatic Speech Recognition system the acoustic and language models were trained with the mixtures of read and spontaneous speech combined in various ratios. In this way the optimal language model and mixture ratio of read and spontaneous corpora were found. The system currently gives a satisfactory peak performance of 18.8% Word Error Rate for a 1:1 ration mixture of read and spontaneous speech, which is comparable with the best word error rates of most of the recognition systems for spontaneous speech available for any language. The systems shows a potential for further improvement and promises to be a nucleus for further work and research in Urdu speech recognition.

Chapter 7

Future Directions

7.1 Size of the training data

The experiments clearly indicated that an increase in the training data resulted in a proportional decrease in the word error rate. This implies that the training data set is not yet saturated (as it was derived from a speech of 140 minutes which contained 114990 phone occurrences of 62 phones only). Therefore, the first thing that needs to be done is to increase the sizes of the read and spontaneous speech corpora. For read speech, this will provide a repetition of the sentence corpus which will result in better triphoneme models. In case of spontaneous speech, the additional data will come from more interviews, and will also enrich the vocabulary of the system from the perspective of trained words. Only after finding the peak of performance (with respect to WER) should more tuning of parameters be done.

7.2 Phonetic Transcription of Speech Corpus

In order to simply the task we relied on the phonemically transcribed lexicon for transcribing the speech corpus. Since neither the corpus nor the lexicon has been completely diacritized this gives rise to two main sources of transcriptions error:

7.2.1 Diacritization Errors

These errors result from the fact that Urdu relies on diacritics for its short vowels. As the native speaker of Urdu can easily guess the diacritics from the context therefore, the written text is hardly ever fully diacritized. While this has no negative impact on the readability of the material it may affect the performance of the training system. A word for example اُس transcribed in the corpus as اس may be arbitrarily mapped onto اِس or اُس which are both valid entries in the lexicon. This means that such words must be disambiguated to accomplish an error free training of short vowels. However, to achieve this goal, two things must be done. The training corpus must be fully diacritized (as much as is required for disambiguation) and secondly, all the diacritized entries must be made available in the lexicon with proper phonetic transcription. This can be largely handled by the Letter to Sound mapping utility.

7.2.2 Mispronunciation Errors

These types of errors result from the mispronunciation of phones done habitually or by mistake by the speakers. The reason may even be the accent of the speaker or some disability. In any case the sounds actually uttered will not match the phone sequence given in the lexicon. If such a problem is either habitual or a result of some disability or habit, the speaker may be asked to repeat it. Otherwise it may be removed in the quality assurance phase. However, if it represents some valid (or widespread) version of the word (or sound), then it must be detected and added to the lexicon as an alternate phonetic transcription. This may require careful analysis of the spoken data while the transcription is done.

7.3 Large Vocabulary Transcribed Corpus for Language Model

The experiments clearly indicated the requirement of a large vocabulary speech corpus that can be used to build the language model. During our experiments the language model was simply being generated from approximately 140 minutes (2 hours 20 minutes) of transcribed speech data (with a vocabulary of 6693 unique words and 31135 tokens only). This is a poor and insufficient representation for the spontaneous and read speech of Urdu. Therefore, one of the primary goals of the future work on Spontaneous Urdu Speech Recognition should be the development of a large vocabulary transcribed spontaneous speech corpus, so that the language specific constructs and grammar of Urdu can be sufficiently represented in the language model.

7.4 Boot Strapping with Hand segmented Speech Corpus

We have used embedded training throughout the training process. However, it is advisable to start with at least some hand segmented data to boot strap the system and then use embedded training in the remaining training process. Embedded training was used to save time as the ratio between the duration of recorded speech to the time it needs to be hand segmented and labeled is roughly 1:400 [1]. However, for the project this process can be followed for some initial data which may improve the acoustic models.

7.5 New Romanization Scheme for Roman to Urdu conversion

In Urdu the letter to phoneme mapping in many cases is many-to-one. Therefore if the CISAMPA based romanization is used we face the problem of homophones when roman to Urdu conversion is done. For example زَن and ظن are two different words in Urdu which map onto the same phonetic construction /z ∂ n/ ([Z A N] in CISAMPA). Since in my romanization scheme the words are romanized by converting them to CISAMPA first that is the romanization of both these words will be [Z A N], in CISAMPA. Similarly letters like ط، ت،ة will be mapped onto [T_D] and ز، ذ، ظ، ض onto [Z] etc. The problem therefore occurs when such words are converted back to Urdu and a stochastic method like the N-gram language model has to be used to weigh the contextual probability of زَن vs. ظن in the roman to Urdu converter. However, there is an easier solution to this problem by which it can be automatically solved by the language model based prior probability *P(W)* used by the ASR in the decode phase. That can be accomplished by modifying the romanization scheme to differentiate these two words by giving them different roman representations.

This requires a grapheme rather than phoneme based romanization. An alternate representation was developed for the romanization as shown in Appendix B. This scheme maps every Urdu letter to a new ASCII symbol. While it results in a lack of readability in the romanized text, it solves the homophone problem.

The work however, should be continued to develop a more readable romanization scheme and a better language model to disambiguate the Homophones. However, the actual solution to will be the introduction of Unicode support in Sphinx. In that case there will no longer be a requirement for a romanization scheme and we can deal with Urdu and other Unicode script based languages as easily as plain text.

7.6 Speaker Independence and Training for Telephone Speech

In order to make this problem a feasible one to solve in a year's duration the simplifying assumption of speaker specific system was made. Otherwise, the job of corpus development coupled with the interviews and transcription etc. as explained in the methodology section, would have been too large a task. However, now that the system has been successfully trained for a single speaker and the basis and procedures for all the work have been established it would no longer be a hard job (although it is certainly time consuming) to train the system with data from multiple speakers. Secondly, according to the requirement of the project being done by CRULP, the next phase would be to obtain the training data simultaneously on microphone and telephone channels so that the system should be trained and tested for spontaneous telephone based speech. The primary changes would be the channel characteristics introduced by the wideband phone channel and the features introduced by the VoIP protocol required for the digitization of the speech.

Chapter 8

Bibliography

[1] D. Jurafsky and J. Martin, *Speech and language processing*. Prentice Hall, 2008.

[2] P. Ladefoged, "A course in phonetics," 1982.

[3] K. Johnson, E. Strand, and M. D'Imperio, "Auditory–visual integration of talker gender in vowel perception," *Journal of Phonetics*, vol. 27, no. 4, pp. 359–384, 1999.

[4] P. Ladefoged, "THE DESCRIPTION OF TONGUE SHAPES," *Dynamic Aspects of Speech Production: Current Results, Emerging Problems, and New Instrumentation*, p. 209, 1977.

[5] "Ess23-voicesystems." http://www.mssociety.org.uk.

[6] L. Rabiner and R. Schafer, *Digital processing of speech signals*. Prentice-hall Englewood Cliffs, NJ, 1978.

[7] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.

[8] "Cmusphinx: The carnegie mellon sphinx project." http://cmusphinx.sourceforge.net/html/-cmusphinx.php.

[9] "Speech at cmu." http://www.speech.cs.cmu.edu/.

[10] "The hieroglyphs sphinx documentation and tutorial." http://www-2.cs.cmu.edu/~archan/-documentation/sphinxDocDraft3.pdf.

[11] "Sphinx 3, s 3.x decoder." http://cmusphinx.sourceforge.net/sphinx3/doc/-s3_description.html.

[12] "Sphinx version comparison." http://cmusphinx.sourceforge.net/html/compare.php.

[13] "Sphinx-4 - a speech recognizer written entirely in the java (tm) programming language." http://cmusphinx.sourceforge.net/sphinx4/.

[14] “Cmu slm toolkit.” http://www.speech.cs.cmu.edu/SLM_info.html.

[15] “Ethnologue website.” http://www.ethnologue.com.

[16] S.Hussain, *Phonetic Correlates of Lexical Stress in Urdu*. PhD thesis, Northwestern University, Evanston, USA, 1997.

[17] S. Hussain, “Letter to Sound Rules for Urdu Text to Speech System,” 2004. Proceedings of Workshop on “Computational Approaches to Arabic Script-based Languages”, COLING 2004, Geneva, Switzerland.

[18] S. Hussain, “www.LICT4D.aisa/Fonts/ Nafees_Nastalique,” 2003. Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center, Singapore.

[19] M. Afzal and S. Hussain, “Urdu Computing Standards: Development of Urdu Zabta Takhti (UZT 1.01),” 2001. Proceedings of IEEE International Multi-topic Conference, Lahore, Pakistan.

[20] S. T. Abate, W. Menzel, and B. Tafila, “An amharic speech corpus for large vocabulary continuous speech recognition,” ISCA, 2005. Ninth European Conference on Speech Communication and Technology.

[21] B. Bozkurt, O. Ozturk, and T. Dutoit, “Text design for tts speech corpus building using a modified greedy selection,” ISCA, 2003. Eighth European Conference on Speech Communication and Technology.

[22] V. Chourasia, K. Samudravijaya, and M. Chandwani, “Phonetically rich hindi sentence corpus for creation of speech database,” *Proc. O-COCOSDA*, p. 132–137, 2005.

[23] P. A. Heeman, “The american english sala-ii data collection,” 2004. Proceedings LREC.

[24] L. Villaseñor-Pineda, M. Montes-y Gomez, D. Vaufreydaz, and J. F. Serignat, “Experiments on the construction of a phonetically balanced corpus from the web,” *Lecture notes in computer science*, pp. 416–419, 2004.

[25] V. Digalakis, D. Oikonomidis, D. Pratsolis, N. Tsourakis, C. Vosnidis, N. Chatzichrisafis, and V. Diakoloukas, “Large vocabulary continuous speech recognition in greek: Corpus and an

automatic dictation system," ISCA, 2003. Eighth European Conference on Speech Communication and Technology.

[26] D. Binnenpoorte, C. Cucchiarini, H. Strik, and L. Boves, "Improving automatic phonetic transcription of spontaneous speech through variant-based pronunciation variation modelling," p. 681–684, 2004. Proceedings of the International Conference on Language Resources and Evaluation (LREC).

[27] A. L. Ronzhin, R. M. Yusupov, I. V. Li, and A. B. Leontieva, "Survey of russian speech recognition systems,"

[28] G. Anumanchipalli, R. Chitturi, S. Joshi, R. Kumar, S. P. Singh, R. N. V. Sitaram, and S. P. Kishore, "Development of indian language speech databases for large vocabulary speech recognition systems,"

[29] A. Li, F. Zheng, W. Byrne, P. Fung, T. Kamm, Y. Liu, Z. Song, U. Ruhi, V. Venkataramani, and X. X. Chen, "Cass: A phonetically transcribed corpus of mandarin spontaneous speech," ISCA, 2000. Sixth International Conference on Spoken Language Processing.

[30] G. Raškinis, "Building medium-vocabulary isolated-word lithuanian hmm speech recognition system," *Informatica*, vol. 14, no. 1, pp. 75–84, 2003.

[31] Y. C. Yio, M. S. Liang, Y. C. Chiang, and R. Y. Lyu, "Biphone-rich versus triphone-rich: a comparison of speech corpora in automatic speech recognition," pp. 194–197, 2005. Cellular Neural Networks and Their Applications, 2005 9th International Workshop on.

[32] T. Slobada and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4, 1996.

[33] M. Goto, K. Itou, and S. Hayamizu, "A real-time filled pause detection system for spontaneous speech recognition," in *Sixth European Conference on Speech Communication and Technology*, ISCA, 1999.

[34] J. Duchateau, T. Laureys, and P. Wambacq, “Adding robustness to language models for spontaneous speech recognition,” in *COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, ISCA, 2004.

[35] P. Beyerlein, X. Aubert, R. Haeb-Umbach, M. Harris, D. Klakow, A. Wendemuth, S. Molau, M. Pitz, and A. Sixtus, “The Philips/RWTH system for transcription of broadcast news,” in *Broadcast News Workshop’99 Proceedings*, p. 151, Morgan Kaufmann, 1999.

[36] J. Gauvain, L. Lamel, G. Adda, and M. Jardino, “Recent advances in transcribing television and radio broadcasts,” in *Sixth European Conference on Speech Communication and Technology*, ISCA, 1999.

[37] H. Yu, T. Tomokiyo, Z. Wang, and A. Waibel, “New developments in automatic meeting transcription,” in *Sixth International Conference on Spoken Language Processing*, ISCA, 2000.

[38] T. Hori, D. Willett, and Y. Minami, “Paraphrasing spontaneous speech using weighted finite-state transducers,” in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, ISCA, 2003.

[39] V. Rangarajan and S. Narayanan, “Analysis of disfluent repetitions in spontaneous speech recognition,” *Proc. EUSIPCO 2006*.

[40] F. Wessel and H. Ney, “Unsupervised training of acoustic models for large vocabulary continuous speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 23–31, 2005.

[41] H. Soltau, G. Saon, B. Kingsbury, J. Kuo, L. Mangu, D. Povey, and G. Zweig, “The IBM 2006 Gale Arabic ASR System,”

[42] J. Nedel, R. Singh, and R. Stern, “Automatic Subword Unit Refinement for Spontaneous Speech Recognition Via Phone Splitting,” in *Sixth International Conference on Spoken Language Processing*, ISCA, 2000.

[43] “Center for research in urdu language processing.” http://www.crulp.org/.

[44] “Linguistic data consortium, ldc2007s03 - arl urdu speech database, training data.” http://-www.ldc.upenn.edu/CatalogEntry.jsp?catalogId=LDC2007S03.

[45] “Sampa computer readable phonetic alphabet.” www.phon.ucl.ac.uk/home/sampa/.

[46] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, “Introduction to algorithms,” 2001.

[47] H. Traunm\"uller, “Coarticulatory effects of consonants on vowels and their reflection in perception,” in *The Proceedings from the XIIth Swedish Phonetics Conference*, pp. 141–144, Citeseer, 1999.

[48] “Praat: doing phonetics by computer.” http://www.fon.hum.uva.nl/praat/.

[49] “X-sampa.” http://coral.lili.uni-bielefeld.de/langdoc/EGA/Formats/Sampa/sampa.html.

[50] “Arpabet and the timit alphabet.” http://www.laps.ufpa.br/aldebaro/papers/-ak_arpabet01.pdf.

[51] “Sphinx knowledge base tool.” http://www.speech.cs.cmu.edu/tools/lmtool-adv.html.

[52] “Sox - sound exchange | homepage.” http://sox.sourceforge.net/.

[53] “Ipa to sampa conversion table.” http://www.crulp.org/software/langproc/-IPA_to_SAMPA.htm, http://www.crulp.org/Downloads/langproc/Urdu_IPA_to_Sampa.pdf.

[54] “Robust group tutorial.” http://www.speech.cs.cmu.edu/sphinx/tutorial.html#app1.

[55] “How to use models from sphinxtrain in sphinx-4.” http://www.speech.cs.cmu.edu/sphinx/-sphinx4/sphinx4-1.0beta/doc/UsingSphinxTrainModels.html.

[56] “Sphinx-4 frequently asked questions.” http://www.speech.cs.cmu.edu/sphinx/sphinx4/-sphinx4-1.0beta/doc/Sphinx4-faq.html#learn_jsgf.

[57] “Instruction set for training.” http://www.speech.cs.cmu.edu/sphinxman/fr4.html.

Appendix A

IPA to SAMPA and Case Insensitive SAMPA (CISAMPA) Mapping

#	IPA	SAMPA	CISAMPA
1	p	p	P
2	pʰ	p_h	P_H
3	b	b	B
4	bʰ	b_h	B_H
5	m	m	M
6	mʰ	m_h	M_H
7	t̪	t_d	T_D
8	t̪ʰ	t_d_h	T_D_H
9	d̪	d_d	D_D
10	d̪ʰ	d_d_h	D_D_H
11	t	t’	TT
12	tʰ	t’_h	TT_H
13	d	d’	DD
14	dʰ	d’_h	DD_H
15	n	N	N
16	nʰ	n_h	N_H
17	k	K	K
18	kʰ	k_h	K_H
19	g	G	G
20	gʰ	g_h	G_H
21	ŋ	N	NG
22	ŋʰ	N_h	NG_H
23	q	Q	Q
24	ʔ	?	Y
25	f	F	F
26	v	V	V
27	s	S	S
28	z	Z	Z
29	ʃ	S	SH
30	ʒ	Z	ZZ
31	χ	X	X
32	ɣ	7	7
33	h	H	H
34	l	L	L
35	lʰ	l_h	L_H

36	r	R	R
37	rʰ	r_h	R_H
38	ɽ	r’	RR
39	ɽʰ	r’_h	RR_H
40	j	j	J
41	jʰ	j_h	J_H
42	ʧ	t_S	T_SH
43	ʧʰ	t_S_h	T_SH_h
44	ʤ	d_Z	D_ZZ
45	ʤʰ	d_Z_h	D_ZZ_h
46	u	u	UU
47	ũ	u~	UUN
48	o	o	OO
49	õ	o~	OON
50	ɔ	O	O
51	ɔ̃	O~	ON
52	ɑ	A	AA
53	ɑ̃	A~	AAN
54	i	i	II
55	ĩ	i~	IIN
56	e	e	AE
57	ẽ	e~	AEN
58	ɛ	E	E
59	æ	{	AY
60	æ̃	{~	AYN
61	ɪ	I	I
62	ʊ	U	U
63	ə	@	A

▨ Rare or no longer in use

References: [45], [53]

Appendix B

Grapheme to Roman Transcription

#	Letter	Roman
1	ے	Y2
2	ی	Y1
3	ة	U4
4	ۂ	U3
5	ہ	U2
6	ھ	H2
7	ں	N1
8	ڑ	R1
9	ڈ	D1
10	ٹ	T2
11	ء	U1
12	ي	Y
13	و	V
14	ه	H1
15	ن	N
16	م	M
17	ل	L
18	گ	G
19	ک	K
20	ق	Q
21	ف	F
22	غ	G1
23	ع	E
24	ظ	Z4
25	ط	T1
26	ض	Z3
27	ص	S2
28	ش	X

29	س	S
30	ژ	Z2
31	ز	Z
32	ر	R
33	ذ	Z1
34	د	D
35	خ	K1
36	ح	H
37	چ	C
38	ج	J
39	ث	S1
40	ت	T
41	پ	P
42	ب	B
43	◌ٰ	W9
44	أ	A2
45	آ	A1
46	ا	A
47	ئ◌	W8
48	ؤ	V1
49	ء	H2
50	—	W7
51	◌ّ	W6
52	◌ِ	W5
53	◌ُ	W4
54	◌َ	W3
55	◌ً	W2
56	◌ٔ	W1
57	U+200D	U5
58	U+200C	U6
59		U7

Appendix C

List of Sentences (Complete)[4]

نیلم نے سالگرہ پر ہیڈ سیسموگراف اسود قریشی کے ماتھے پر اینٹھن اور غم کی آتشیں رو محسوس کی

حاجی مجاہد بلگرامی مخزن اور غزوہ کے ایک ارب قارئین میں انتہائی صادق اور جینوئن قاری تھے

سامعین انفارمیشن کی گھن گرج سنیں تو ویزے کی رپورٹ میں پوشیدہ ایک محدود ایل وی ڈومیسٹک پیکج ہے

کمیونسٹ لوگوں نے تنگ ہونے کے باوجود کئی شعبوں میں تندہی سے اپنے کیریئر کو مزین کر لیا

ٹرانسفارمر پر مڈنائٹ میں شاہین گدھ اور عقاب سمیت چیسٹ کے بل سرعام سینکڑوں برڈ بیٹھتے ہیں

کڑوے قہوے کا شیدائی اصغر کاشمیری باغبانی سیکھنے والا پانچواں منیجر ہے

پٹھوں اور سر کا حادثاتی درد بام سے فوراً سلجھائیں پھر میٹھے اور کولیسٹرول سے بچیں

کیولری ڈیم کی موزونیت سمجھانے اور ادارے کے ارکان کی تربیتی خدمات کے لیے حبیب نے تحریک چلائی

اقتصادی معاملات کی تفہیم اور فرماں بردار نوجوانوں کی بریفنگ گفتگو میں سرفہرست لکھیں

سجاد ریسکیو معلومات کھنگالنے کے بعد جانے کیوں بلاسوچے اچانک سوئمنگ پر راغب ہے

حامد سب ڈیلروں کے سامنے ان ویڈیوز کے بدلے میں اپنے اعزہ کو ایک ماچس بھی نہیں بخشے گا

تم ڈائیلوشن کے عمل سے بچنے کے لیے نتھنوں سے گاؤ کیونکہ جبڑا سوجنے سے تلخی اور سوجن ہو گی

شجر کی بیرونی شاخ پر گیارہ ڈوڈیاں توازن کی درست مثال ہیں جو سنگھاسن بیضے کے مترادف سمجھی جاتی ہیں

ڈیڈ پچز پر کیچ میچ کا رخ پھیریں تو نتیجے میں بیشتر لوگ خوشی سے رو کر بھی قیامت ڈھا دیتے ہیں

مملکتیں بجھی اور بے ڈھب موجوں یا صیہونی ڈاج سے رعشہ زدہ نیشن کو ٹرانسپیرنسی نہیں دے سکتی

ریاست کے چوالیس تعلقوں میں صیغہ بسائیں تو روہی اور متھرانی زمین مزارعوں کو پسند ہے

سرویئر نے انستھیزیا اور مالیخولیا میں پائیریسی سے بچاؤ کے تھیم میں خود کو منوا لیا ہے

زیست کے چنگل کے اس فیز میں جیون ایک افلاس میں پھنسی پرنم آنکھ کی طرح ہوتا ہے

برسراقتدار اہل دانش سے قسم یا حلف کی گٹھڑی اٹھوا لو تو فرض اولی چھپ جاتا ہے

[4] Developed by the team of researchers of the *Center for Research in Urdu Language Processing* (CRULP), working on the project entitled *Telephone-based Speech Interfaces for Access to Information by Non-literate Users,* from the phonetically rich word list generated by the greedy set cover algorithm

روضہ کی دیسی ساختہ فیبرکس کا ذریعہ انٹرفیس میں درج ہے
مونا اس سفر میں الجھ کر سنبھلی اور ٹھہرنے میں کامیاب رہی
اخوندزادہ نے ساتویں سنچری نئے طرزعمل اور استقلال سے بنائی اور اؤٹ ہو گیا
ایکسیئن نے اسی موشن میں ایمبیسی کی تحویل میں گھسنے کی کوشش کی
ہوشربا کڑوی تبدیلیاں عیاں اور سازباز کی جڑیں ملغوبہ انداز کی ہیں
ننانوے ملین متاثرین سے پوچھے بغیر آپ کا تعاون معاف نہیں ہو سکتا
وفاق میں منظم آتشزدگی پر چھپانوے بااصول ملازمین کا استفسار بالکل چاہیے
حملوں سے مجید صاحب کا بیس بیڈروم کا گھر اتھل پتھل ہو گیا ہے
کانگرس اسی کوڈ میں انسان کی موزوں قدروقیمت ظاہر کرتی ہے
عابد اسے جھڑک کر بھیجے اور ریٹائرمنٹ کے بعد اس کی کفالت سے مستغنی ہو جائے
بپھرے ہوے لوگوں کو چھوئیں اور ننگے الفاظ سے سنگباری نہ کریں
ہمارا پیغام ہے نماز پڑھو کاجو کھاؤ اور اپنی عمریں علم کے لیے صرف کرو
خانقاہوں میں زبوں حال نشئی غلیل چوری کرنے کے لیے چھڑیاں اٹھائے کھڑے ہیں
ویسٹ فن لینڈ میں باؤلر جھٹکے سے زیب کو کچوکے لگاتا رہا اور رنز بٹورتا رہا
کانگریس حکومت یہ پورا قومیائے تو خدوخال ابھریں اور پلوں میں دھبے مٹیں
چوزوں کا جھلس کر بیضوی لاشیں بن جانا صلاحیت یا رولز کی کوتاہی ہے
نیم بحرانی مورخین کی تجرد میں قید ایک خواستگار کی توہین ہے
ایڈونچر میں گیر لگانا گھوڑے دوڑانے سے آسان اور سیف سمجھو
زیڈ گھمن نے دو مرلہ پر سنگھاڑے اگائے تو بچہ کھلکھلا کر پھدکتا ہوا وہاں آ گیا
شہباز کو ہیجڑوں کے شوکیسوں کی پہچان ہے
یورپی نوبل شیئرنگ ایمبولینس کے نشر کے لیے یہ پیچیدہ ساعت ہے
بوڑھوں کے لیے نل کی دیکھ بھال ایک تذلیل ہے اور ناقابل افورڈ ہے
میرے مطابق یہ فیاضی ٹھیک نہیں آپ غیر حاضری کی پروا کریں اور تین بجے ریڑھی بھیج دیں
میچ میں دھویں سے راجے کی کھال بہت بری ہو گئی ہے
شہید کے اقربا نچلی کوئین گیلری میں ہیں اور پارلیمنٹرین گمشدہ تمغہ ڈھونڈھ رہے ہیں
شکار کے لیے ارمغان سے پوچھوں تو علیحدہ چونچلے ہیں
مشرق کے کیسز میں ثلث کی یہ توجیہ بوگس میں چھپ رہی ہے
شاخوں کی گروتھ اچھی ہے اور چوکھٹ پر پنچھی خوفناک دہشت میں مبتلا ہے
لوک سبھا کے پرچم تلے مجیب نے گلوبل اتفاق کی مدافعت پر بات کی
قلزم میں پلوں سے تھوڑا یا دو گھونٹ پانی ڈالنے سے بوجھ نہیں پڑتا
شیخ ذوالفقار کی کئی شاخیں پبلشنگ میں پیشرووٴں سے ناجائز فیض یاب ہیں

پوچھنے پر مغرور سٹوڈنٹس نے کوارڈینیٹر اور امپائر کو ہراساں کیا
رینگتے ہوئے اس مہم میں سوشلزم کے لیے بغاوت کا علم بلند کیا
مواضعات میں بنگالی مندری کا تھیوا اور سوئچ بیچنے جایا کرتے ہیں
پھونک سے بھینس کے تھنوں کی سرخی قدرے رخصت ہوئی
صحبتوں میں گالف اور آنکھ مچولی جوق در جوق کھیلتے ہیں
وعدے وعید کے بعد شوکت کی نمک خواری رائگاں جانا پرلطف ہے
میت کو روڈ سے قبرستان بھجوائیں اور درود پڑھنے کی کوشش کریں
دوغلا شخص جونیجو ہے تو سزا کے لیے پھانسی نامزد کرو
گردش کے اوپر گائنی وارڈ میں ریسورس ٹریبیونل ہے
تبسم جمخانہ میں کیموتھراپی کی توضیحات پر بحثتا ہے
نعیم ڈار برش یا انگلی سے تھوتھا جگ میں گھول کر لائے
تاجور مصدق نے سرحد میں اناسیواں نیشنل مرغبانی مینجمنٹ پراجیکٹ منعقد کیا
باران رحمت میں آمدورفت سے بوڑھی خاتون تھک گئی
اوجھڑی کیمپ کے لیے خلائی نگہداشت معروضی سمجھیں
اختر نے جنم بھومی اور دائیں آماجگاہ سے ملحقہ جزو بنایا
مجھے زودحس اور اوچھے اوصاف بخشیں اور فضول حزن سے بچائیں
آپ دونوں بجنوری صاحب کے ساتھ چلیں اور غزلیات کی بعینہ تصدیق کریں
دارلخلافہ میں ویڈنگ ایڈوائزری بورڈ معرض وجود میں آیا
زلزلے نے طرفین میں ساری گاڑیاں پسماندہ بنائیں
مغز لیچی وغیرہ سے پرہیز اور روزمرہ میں ٹھنڈا جادوئی کنواں استعمال کریں
صوفیہ نے دائیں گھوم کر منجھے بھگوڑوں کی ڈیویلپمنٹ پر شیم کہا
ایڈیشنل ججوں کی طویل ادائیگیوں سے سوئس بینک منافعوں کے اگلے اعدادوشمار کے جدول ریلیز کر دیے گئے
تنخواہ کے بعد جیب اور جسم تھر تھر کرنا سجھائی نہیں دیتا
مداحوں نے روشنی میں کھلاڑی کو اچھے کھیل پر گفٹ دیا
انوسٹمنٹ پر مدون ضخیم دستاویزات مشاورت کے لیے ساؤتھ لیبارٹری میں ہیں
بانو نے بیساکھی پر ہندوؤں کو تحائف دیے جو بہتر ذوق کی درخشاں قدسی شروعات ہے
کشف ایک پرسوز شعاع ہے جو خود شناسی اور مطالعہ کا ثمر ہے
قادیانی جالندھری اور کشمیری اس لائحہ عمل میں موردالزام ہیں
دیہاتیوں کے دھڑے کی اکٹھی چار بھینسیں ڈوب گئیں
روزہ اویئرنس کی افزائش کرتا ہے اور تعلیم کا چراغ ہے

پیشوا نے ملیریا اور پھیپھڑے کے مریضوں کو ڈسچارج کرا دیا

تحصیلدار صاحب بنگلے اور کوٹھیاں کھلوائیں اس کے سوا کیوڑہ نما جوئیں نہیں جا سکتیں

مورچے میں ایس تارڑ بخوبی اپنے موقف اور پالیسی پر چلے اور صرف نام کے تنازعے دکھائی دیے

راہبر استاد اور مورخ بلا شبہ بحیثیت مجموعی گلستان کے مصداق گزرے

دیکھو سوزش نے شیر کا کیا حلیہ بنا دیا ہے بلکہ اس کی دھاڑ سے لومڑ اور چوزے بھی چیخنے لگے

انہوں نے ٹیچنگ ہاؤسنگ اور ہیلتھ میں تنقیص دیکھی اور نااہلیت پر بخدمت فنانس وزیر درخواست لکھی

رقیہ نے زرخیز زمین میں ہرادھنیا بویا اور پھر تفاوت ختم کرنے کے لیے شہزادی کی نذر کیا

مغیث نے یقیناً کافی تذبذب سے بلاّ اٹھایا اور کہا آگے چلو کھیلیں

سکولوں میں یگانگت متعارف کروانے کے لیے طالب علم خود جوش و خروش سے اٹھیں گے

اخبارنویسوں نے گلیشئر سے پھسل کر بیلچہ گرنے کے واقعے کو اچھال دیا

بھوت بلا سے وحشت معمہ ہے اور پریوں کے چہروں کی شوخی ایک فوبیا ہے

کوئی بلدیاتی چیلہ مچھر سے دفاع کی پاسداری کی ہنگامہ خیز پیشکش کرے

سوویں صفحے کے نچلے حصے پر قرضے کے اوقات دیکھیں

معصوم بچہ خفگی سے رخساروں پر پنکھڑی پھیر رہا ہے

امیگریشن کے لیے ٹوفل پروگرام کی ازسرنو ترویج میں بالواسطہ غور و فکر موجزن ہے

کسی درسگاہ میں خوارزمی کی کتابیں کتابچے اور باب ڈگری پر فوقیت رکھتے ہیں

اطالوی مہمانان اور سیاح مجلس میں ٹیبلو اور کامیڈی اجزا دیکھتے رہے

ندوی انشورنس کے انٹلیکچوئل نے کہا تفرقہ اور میثاق کی مذہبی حدود میں رہیں

محافظوں اور عہدیداروں کے نرغے میں پریس پہنچوں تو اشرف کے بدلے ہوے اور زاید فیصلوں پر نظرثانی کراؤں گا

اخروٹ کے چھلکے اور انار کی گٹھلیاں دودھ میں بھگوئیں اور اسہال کی افزودگی میں بچی کو دیں

دلشاد جکھرانی نے برعظیم میں پروسیسنگ فارمولے کے افادہ کو جوش میں مسترد کیا

بچو جامعہ کے اسٹورز سے ارشد علوی کے لیے چاول یخنی اور چلغوزے لاؤ

دلبرداشتہ خواتین چونگی آرڈیننس کے بعد سیل پوائنٹ پر مرچوں اور دوسری لائن میں مصروف ہیں

نیلوفر کو کوہستان میں دشمن کی فوجی جدوجہد اور سیل رہزنی پر تشویش ہے

کوریوگرافر ہذیانی ایتھرنیٹ سے بچاؤ کے لیے ہسپانوی حکومت کی مہربانی اعانت اور وسائل سے بلاخوف کھیلے

حسین عقیل رندھاوا نے اصل میں دریچوں میں ایندھن بکھیرنے کی وجہ سے ٹریولرز سیفٹی انسولیشن سے مذاکرات کیے

سوختہ گرداں ادیبوں کی ملفوظات میں چاشنی ہنوز اشاعت سے محروم ہے

مادھو نے لیڈز اور کاندھلوی جونیئر ناشتہ والوں کے خلاف پردھان کو زیرعنوان گھوسٹ قرارداد دی
گوٹھوں کی کھدائی میں مصحفی کا کھوکھلا آسرا غیرقانونی تیروں کی کثرت سے کچلا گیا
قصوروار صیاد کا داغدار نوحہ جعلی غنودگی سے ہرگز چونکا تھا
تنظیم آزادی کے جلسے میں مخالفوں کے تاثرات کا سدباب کیا گیا
اپنے آنگن کے باغیوں سے ہتھیار کے بجائے ہوشمندی سے نمٹیں
ایوان میں جاگیرداروں اور وڈیروں نے پرائیوٹائزیشن کے معاہدے اور معاوضوں پر بات کی
ایجوکیشن اور نشرواشاعت میں برطانوی اور آسٹریلوی مدمقابل ہیں
نوآبادیاتی کونسل کے ایگزیکٹو ایری لنگا نے ٹیکنالوجی کے قرضوں کو برقرار رکھا
ورلڈکپ نے اقوام کا افسوسناک تناؤ اور انگیخت ہیچ کیا اور دنیا میں شگفتہ کلچر بیدار کیا
ریحانہ نے رقاصاؤں سے بےجا بحثیں شروع کیں
رباب طفیل کے غیردانشمندانہ فیصلے نے دہندگان کے توقعات اور جذبات کو پژمردہ کر دیا
بدقسمتی سے شگاف تین فٹ گولائی میں تبدیل ہو گیا ہے لہذا اسے بند کرواؤ
شواہد کے مطابق انفلوئنزا کے لیے ہومیوپیتھک کی تدابیر بھونڈی اور بس جونہی سی ہیں
پچاسویں حاشیے کی لینتھ تیسرے حاشیے سے غیر شعوری مماثل ہے
ہندوستانی ہوزری کے انجنیئر اور تاجر چوروں میں پھنسے ہیں
یوریا نے چھاؤنی میں لونگ کے باغوں کو بچایا
کبیر بڑھئی کا شہرہ اور فیم ہی اس کی معیشت کا بھروسہ ہے
کیا پروفیسر طیب اور لائبریرین نے یونیورسٹی میں منیجنگ ایکویٹی کا عمیق معائنہ کیا
بعض بارونق خیمے پھیل کر قدوقامت میں بڑھ گئے
ہر اناڑی اور راہگیر ایک سپاسنامہ لے کر معاشیات کا معلم سمجھنے لگا ہے
چودھرائن نے بوڑھے ماچھی رساؤ کو افضل کی عیادت کے لیے بھیجنے کا سوچا
ازبک صبروتحمل سے ایک برس کے قیام میں اچانک پرسکون اور مالدار بن گئے
عالیشان وزارت نے حفیظ خیلوی کو تساہل سے زنگ آلود بنا دیا
منفرد ناول میں دھینگا مشتی اور تشدد کے برعکس منشور اور مقصود سے معجزے کی پیدائش ہوتی ہے
خواہش دلوں کو خودسر اور جنگجو بنا دیتی ہے
جمشید بےاختیار ویسے ہی آدھے کیچڑ میں پہنچ گیا حالانکہ ہمارے ہجوم میں تھا
چمن میں منظور شدہ مینوفیکچرز کا انکلوژر نوعی ذریعے سے منہدم کیا گیا
طاغوتی مواصلات کا معاندانہ طرزفکر مسلسل ڈیزاسٹر اور اجاڑ دیکھ رہا ہے
دیباچے میں ہجرت اور ایڈز جیسی بھیانک اور ڈراؤنی مرض کی ایلیمنٹری روئیداد درج ہے
صبغت انیق فرخی نے ڈیکلریشن نمبر موقر کو دیا اور زخمی کی دلجوئی کی

روسی انٹیلیجینس میخائل وولر کے متعلق عملا رقیب تھی

کروڑوں ڈیزائنروں نے سائبان کی بیسمنٹ میں بغدادی سازوسامان کی آڑ میں زیورات خریدے

لابنگ بیورو نے غیررسمی گرمجوشی سے کہا ڈونرز کو آزمائیں

محققین نے متفقہ باؤنڈری کی تاریخ نظروں میں رکھی

باورچی اور ٹیکنیشنز چلچلاتی دھوپ میں ادویات کے لیے پریشاں ہیں

والدین کی محبت فیملی کے لیے تقویت کا موجب ہے

آج گورنر صاحب غربت اور تنگدستی کے باعث اپیلیں کرنے والوں کو رعائتیں دیں گے

مولوی منیب بخاری عجلت میں قیوم کے ساتھ گورنمنٹ آفیشلز سے ملے

شاہد صاحب کسی دلخراش اور مہیب تنزلی کو قبول نہیں کر سکتے تھے

اگر لفظی دانشور کھلواڑ نہ کریں تو شہریوں کا احتجاج فتحیاب ہو

مجوزہ تشبیہ اور تلمیحات لینگویج اور شعروادب میں وجدان اور صداقت کا باعث ہیں

مولانا خالد شمسی لاہوریات کے عنوان سے تاریخ اور سیاحت پر لکھتے ہیں

موجودہ ایونیو بلوچوں کے زیراستعمال ہے جو گلدستہ اور ہول سیلز کے کاروبار میں مستغرق ہیں

محترمہ پروین مینگل گردش دوراں نامی مدرسہ کے پرنسپل کی دوست ہیں

رفعت لغاری موسمیات اور سیاست کے مسائل سے شدید بیزار ہیں

آرگنائزیشن کے چیئرمین نے بعدازاں شائقین سے اجازت طلب کی

وزیراعلی نے انجمن کے متحدہ کنونشن میں بالخصوص درآمدات کی وقعت بیان کی

گاڑی ریورس کرنے سے لہجے میں شیخی اور کہانی میں زرمبادلہ کا ڈھنڈورا لیفٹیننٹ کا مزاج ہے

لفظیات کے ورژن میں سیونگ کی درخواست جیوری نے تنظیم سے گزارنے کے بعد سردخانے میں رکھی

دستاویزی موومنٹ میں رفیوجی ڈیزرٹ اور میواتی ہجے حقوق کی تھیوری سے کھلیں گے

نائب محرر نے ہتھوڑے کی مدد سے تقریباً سینٹر میں زنگ آلود بیلٹ کو مضروب کیا

اے ضوفشاں کہکشاؤں کے لالچی لوگو افسوس عفو اور صبر ایک کاروبار اور صرف بولنے کی حد تک ہے

سینما گھروں کا رسیا بوڑھا حبشی لیڈر عجیب مسحور آواز رکھتا ہے

جمخانہ کے راولا اینگلو سکول کا اٹھاسی واں صدر حجاج کی ترمیم تجویز اور مشورہ پر سوچنے لگا ہے

ایسے عہدے بزرگ صحافیوں اور لکھنے والوں کے لیے دکھاوے کی کھیر اور زہریلی ترقیوں کی سوغات ہیں

میزبان اس موقع پر پہلو کے کمرے میں گھسے اور متاخرین سے پورے مقابلے کے لیے کہا

غیر سرکاری انفورسمنٹ فورس کی شمولیت چھوڑیں اور کھوسٹ کردار بھیجیں

جسٹس کلیم سے رعیت کے لیے دشوار گیر مشغلہ جلد پوچھیے نیز ذہنی رجوع کے نتائج دیکھیے

عزت مآب رخشندہ کو ڈسٹرکٹ کمیونٹی ڈویژن کی سیر کرائیں اور میتھائل زچگی سنٹر پہنچائیں

آکسیجن کاربن اور یورینیئم کے اوہام پر سوچیے اس بھید پر جید عندیہ دیں

سگے بھائیوں کے کھو جانے پر سلطان زرقاوی ڈھے گیا اور موسلادھار دعائیں کیں

تخلیق کی لو ایک رنجیدہ خاطر موو اور فورسڈ پیشن کا آغاز ہے
پریمئر میں پچھتر گز غیر ملحق قالین زیرقیادت ریوڑ کے لیے نصب تھا
ملت ڈے پر ڈوگر اور میمن کے سامنے فون پر ریلی مقرر کی گئی
سسٹم مرچنٹ کی پرچھائیں ویوز پر پڑی تو قادری کی صلاحیتیں خزاں رسیدہ ہو گئیں
جن کالجز کی فہرستیں ریڈ ہیں وہ کمشنر کو تسلیم ہیں
اجاڑ اشجار کی تعداد اور گیراج کی پیڑی پر اونچی شاخ کے جھونکے ایک تحفہ ہیں
اگرچہ ہیروئن کے کندھوں پر ننھا بچھڑا تھا پر تباہی اور طوفان میں پانچوں بار سنبھل گئی
روزانہ نیوز دورانیے میں انضمام ورک نوشاد سے کتابچہ لکھوا کر دوردراز پہاڑوں میں رکھوا دیتا
کینگرو اور گھاگرا کی پھبتی پر زنخوں کے اوباش کھگہ اور کھچی سے الجھ پڑے
ٹاؤن میں چھاپہ کے لیے انٹیلجنس بھیجوں تو اٹھاون فحش مصنوعات آپ کو ڈبوئیں گی
دیوبند میں گوشہ حسینی کی مراعات میں شاہنواز نے رنگین گوشہ نمایاں کروا دیا
گو شعر و اشعار کے مجموعے ونڈوز سے بیچیں لیکن سوویت بیوروکریسی سے رابطے بڑھانے ہوں گے
میلے کچیلے خام فوم کے سویٹر میں کلراٹھی چقندر باندھیں تو یہی شے خوبصورت ڈھال ہے
آئین پر قضہ فیڈریشنز کے دماغی پہلوؤں کا محور ہے اور واپڈا کے ڈومین سے باہر ہے
خواص کی تلخ رپورتاژ پر بھلا صغیر کو چھیڑنے بھگدڑ مچانے یا ناچنے میں کیا پنہاں ہے
بدصورت بانسری کی انوالومنٹ سے سیلرز کی ہیرو سے گراں تخریب نیچرل تھی
ویڈیو اور ویلڈنگ کے بقایاجات پر گڈریے نے اپنے بہنوئی غنی بلوچ کو مدعو کیا
لائبریری کے مجموعوں میں ایک مجموعہ کروموسوم اور نیورو تفتیش پر ہے
چھہتر اونس سیڈ چون بلین میں ایک بشاش دھاندلی سے لیس شاندار گھاؤ ہے
چھاؤنیاں مرضی کے اویسی نقوش اور جداگانہ پرتعیش روزگار کا مخرج ہیں
گیارہویں جاسوسی وفد نے کارخیر کے منتخبات کو رانجھن چشتی کے خانگی ایڈریس پر بھیجا
آپ چوتھی صوفیانہ سویرا کی چیریٹی لانچنگ نبھائیں تو میں آپ کو محقق بناؤں
دوگانہ سانحے کے بےقصور متوفی منیر گوہر طبعی چہچہاہٹ میں خا صے عزیز تھے
رحیل سندھو نگرانی بڑھائے تو ہم تھل کے مڈھوں میں چمچ جیسے کچھوے ڈھونڈیں
سترھویں گھنٹے میں جلالہ بھرٹ ہابھہ کی جنگوں کے جشن اور شریفین تھیری پر سوچ جا رہی تھی
قدسیہ گھونگٹ میں نبوی راہداری میں کھیلوں کی پرچیاں اور پھول چھانٹ رہی تھی
بدہضمی اور موعود بیماری سے جناب سائیں منور علی بنوری کا رنگ سبز اور بھورا پڑ گیا
میرٹ کوئی شعار یا وصف نہیں بلکہ دغا اور گھناؤنے گردوغبار سے اچاٹ ایک چیز ہے
شون کی تہذیب سے ماخوذ توسیعی چراغاں کی تشہیر میں ایک عبرت اور تحیر غالب ہے
لئیق نے انہیں کہا آپ جائیں شمشاد کو پچھاڑ کر دوڑائیں تو میں باولنگ بڑھاوں
اسجد کے مفلوج ہاتھوں سے لڑھک کر سویڈش پٹاخے کی ڈوریں والیم وولٹیج میں خلل کا باعث ہوئیں

سوئیوں کی جڑواں ٹھیس کا جھگڑا لفظوں کی جھڑپ میں بدل گیا

ایشیائی زون کے ڈیفنس والے علاقوں کی ویمن جذبات میں محویت پر نازاں اور غرارے لگانے والی ہیں

نبویہ پرچھائیاں ربیعہ آفریدی کے لیے نصیحت اور سچ کی وجہ سے فیورٹ ہیں

اژدھام کی جھلک منظوم اور معجل تراشوں میں پیش کرنا ملک انور اور نمف نے سیکھ لیا ہے

نائیٹروجنی ٹری اور نوشیرواں بیراج کی درستگی میں پندرہ گیجٹ بیگز استعمال ہوے

مطبوعہ آفس میں مل کر ٹھٹھر جانے یا لوتھڑا بن کر ہلاک ہونے کا فوجداری تنازع راجیہ سبھا میں ہے

لغو شغل میں ندرت فجور کی جڑ ہے اور ذہنی گھیراؤ کا اجرا ہے

فیاض ویجی باتھو اور بھیم سے ڈائریکٹ لگاؤ بھول کر آڑو اور کلونجی ملحوظ رکھتا ہے

ڈیمو اور کوچوں کا غوغا نو عمر سانولے جیوئش کنیئرڈ سرونٹ کی لیے پریشان کن ہے

حسیں مدح پر ڈویژنل ڈائریکٹر کی طرف سے رعایت بشمول چھٹی تعیشات اور مہربانیوں کی جولانی ہے

تحصیلوں میں ماڈلنگ فلموں کی معنویت خوابیدہ قدرتی تکلف کی بہرحال فراواں صورت ہے

جوں جوں جوے مژگاں کی منقش ٹھاٹھیں اٹھنے کو ہوں روے ارشاد مرغوب اور باغیانہ ہو جاتا ہے

جادوگر کبھی بھی ریچھ اور بھیڑوں کے لیے وڈ کے کنگھے گولڑوی گدڑی میں نہیں رکھتا

بینات نے کس جور کی کفایت کے لیے کثیرالقومی ریزولوشن جمیعت کے وڈیو مائنڈڈ بینچ کو دی

لہذا آپ فاخرہ سے پوچھ کر ریاض کی شادی میں کھلے عام تشریف لائیں

نعوذ باﷲ حشر ایک گنجلک فیتھ ہے بلکہ یہ تو میتوں کی بجاآوری کا دن ہے

اوج اور فرخندہ نے ڈچ جھالر میں مغربی تشخص کا حامل پہلا مہذب براہمن بزنس مین دیکھا

فنڈز یا روپے کی ڈیلیوری بڑھائیں تو سائنسدانوں کا آپریشن اور ترقیات لاینفک ماڈل بنیں

ایرانی قدروں میں بندوق کی روایت ٹھوس نہیں صرف بحثوں اور خبروں کا ایشو ہے

لاشوں کی تدفین کے بعد شاہ جہاں محتاجوں کا نوعی جمگھٹا دیکھ کر نکل گیا

نوچندی بغاوت میں رائیگاں جھانکنے پر مفید اور تاریخی سچائی کا صحیح اعادہ نہیں ہو گا

چمچہ شیرے میں بھگو کر نومولود کے منہ میں رکھیو

آپ ٹرانزسٹر سے نہ الجھیں تو میں جماعت میں ٹورنامنٹس کی کارکردگی پر بولوں

بائیس میگاواٹ کے قریب ویلیو سامنے رکھیے تو ہوشیار پھرتی سمجھ آتی ہے

تصدق روئی اوڑھ کر بڑھا اور عقلمندی سے جدون بنگلہ تک پہنچ گیا

کفار کی سوکھی اور ناآشنا بڈھی کو کلیدی تبلیغ اور دعا کی ضرورت ہے

سوچوں میں کلیرنس اور پاکیزگی ہو تو تفنن سن کر رسوائی نہیں ہوتی

ابو اور چاچو ریجنل ہیڈ ظہور خان کے پاس خیام کا نادر اور شستہ فیچر لے گئے تو وہ بغلگیر ہوا

چھانگا کے تلور کے رنگوں کا ذکر خیرہ کن افیکٹ اور ایک ڈسکوری ہے

نشست کا مقصد وائرس کنٹرول ہے وگرنہ ٹھل اور چھور میں فصل کی مدت اوور ہو گئی ہے
پہلوان بھاگ کر باغ میں گیا چیلنج کیا کہ اگر خوشہ دبایا تو میں کھدر کو جھنجھوڑ ڈالوں گا
مجلے میں مظلام تابش کا منفی رویہ غیرت اور ایجنسیوں کے مطالبے کا ذکر کریو
دسویں اور ہزارویں عہدہ پر حضرت کو کامیاب کانووکیشن کے اخراجات چبھنے کی سمجھ نہیں آئی
بھونچال میں چمڑے اور فولاد کے پنجرے جھوم رہے تھے
منیب ابتدائی آمدنی کو مؤخر کرے تو دو ماہ میں پرفارمنس اور علاج سامنے آئے
سدھو بدظن مکھڑے کے ساتھ مہجور آشنا سے ملا ساحل کی ظریفی کتھا سنائی
ڈھائی صفحوں کا دیباچہ تو ڈبلیو راٹھور کے لیے دوبھر ہے حالانکہ اس فضل کا اہل ہے
استغراق میں پیشرفت کیجیے اور گرہیں کھل جانے کے بعد جلووں کا ادغام چکھیں
ایلویٹ الیون ایونٹ میں خودسوزی کی وجوہات کی ایڈوانسڈ خبریں باخبر سٹوڈیوز میں ہیں
پنجند کی شہرت معاشرے میں ڈیپریشن اور جھجک سے ریلیف کی مظہر ہے
مصور کا ٹیلنٹ جھرنے کا شور اور غیب کا سایہ میں ابھر کر آیا
مشاغل کے دوسرے ڈھانچ پر انگوٹھے کا اشارہ غافل غفور کے لیے ایک چوائس تھا
راسخ واہلہ نے رم جھم آبشار سے پرجوش تیقن کے ساتھ عرضداشت بھیجی
فزیوتھراپسٹ اور پاسبان نے انیس کی بافتوں کو توقع کے مطابق پھیلا دیا تو وہ رویا
گدھا مکھیوں کو دور رکھنے کے لیے ڈھینچوں کی آواز نکال سورما بن کر اودھم مچا رہا تھا
دولہا معا کھانے کے کمرے میں تھریپیس اور شیروانی کی پیشکشیں بیسٹ اور گڈ فقروں میں اڑا رہا تھا
بااثر دھوبی کونسلر ریان کے زیراہتمام یاں ملی سکولز کی تعمیر کے خیرخواہ ہیں
بھدی نفسیاتی جھریاں اور صعوبتیں ڈاٹ ایم پی تھری ڈیجیٹل نظام کے طفیل ہیں
لیزنگ کی خبر شائع ہونے اور پھیلنے سے اسرائیلی فاؤنڈیشن تیار فقرے ڈھول اڑائیں گے
اکھنڈ مشاعرہ میں مرحوم نے دھواں دھار اسلوب سے شعائر کی دھجیاں اڑائیں
بچوں نے سوڈا اور مصالحہ مفلس جوگی کو تحفتا دیے اور نڈر جھینگے بیچ
تمسخر کے دوران جیو کی صحبت میں کوچ حملے سے پھٹ گئی اور عجلت کا فائدہ نہ ہوا
گگلی کے تجربہ اور تشخیص پر ہدایت اور تعریفیں چاہتے ہیں
کمپیوٹرائزڈ ٹوہ کو خیر باد کہو اور آہستہ آہستہ گھنیری جھتیں اور شگوفے اور تلخیاں بھول جاؤ
اچھے اور باذوق مسلم مظفر نے ملیریا اور انجری کو پرکھا اور فوجیوں کو ایمرجنسی بھیج دیا
شہری چڑیل کی ہجویہ آوازیں سن کر چیچن حافظ کے پاس گئے
دوا کے بغیر اٹھائیس گورے ملیشیا کی پہاڑی والی دیوہیکل جیل کی لبالب آلودگی میں روئیں گے
سائیکل کی ابجد سیکھنے کے لیے کاکلیں پونچھ کر ڈام کے تصفیے کے بعد پاٹھک کی صفوں میں جا
بلونگڑے مطمئن ہو کر دیوداس کے پاس پنگھوڑے میں بیسن کی غذائی مٹھل کھا رہے ہیں

پہاڑ سے یکساں میزائل کہکشاں کے طوفاں سے گھائل فائربریگیڈ کو کریش کرنے کے لیے ہمسفر تھے

سیمی نے معمر راہگیر کو جواب دیا کہ میں جو چاہوں بہ رعایت لکھوں

کھمبیوں کے پاس فیشن نے آنکھیں چندھیا دیں کیونکہ مخبری کے منصوبے میں مدعی ٹیڑھا تھا

کرفیو میں گھرے جنگجوؤں میں پھنسے بھنگڑے فائر نگاور چیخیں سنتا عشروں میں پہنچا

مارشل فورم کا پلڑا فیوچر میں ایک اننگز اور سیون سکورز سے صیہونی ڈائینوسارز پر بھاری ہو گا

خلق ذوالجلال داؤ اور قدغن میں الجھی تو اغیار کے پٹھو پرکھے گئے

قضا کی ارزاں چوڑی اور مدھر لہریں نجات کے ستھرے رجحان میں مخل ہیں

مغفور و مدفون نصیب نے کلیجی دھونے کے بعد اچنبھے اور توقف میں بیعت کی

ناخواندگی کے باعث بچیاں سسرال میں سگھڑ کام انجام نہیں دے سکتیں

حکمراں چالیس مویشی کھولنے کے لیے ڈاکوؤں کو جھنڈے اور محفوظ مربعے دے رہے ہیں

عالمی ادب کا نمائندہ اس جگہ تقرری سیو کرنے کے لیے ریڈیو سے ایڈجسٹ کرے گا

رافعہ نے کوچے میں گھس کر فیڈر کے مرکزی گوشے میں دبیز سیسے کی جھلی اور گیسیں دیکھیں

تھیٹر کے رکن شدھی کے ایہام میں ماڑی کے ڈور پر ہجرت کی تمناؤں میں بکھرے

دہلی میں چائے اور پھپھوندی زدہ کانجی کی گانٹھوں سے سوواں گیدڑ بھی ڈھلک گیا

کارخانے میں سویرے کی حبس میں سلفیٹ کی ساخت کا قدآدم اژدھا نما جیٹ اڑایا جا سکتا ہے

ہیلو ہائے رکھ رکھاؤ کا ایک ساغر ہیں اور سوری ایک دعائیہ حوصلہ ہے

اداکارائیں ریٹائرڈ میئر یونی ڈیف کے اخبار کی خوشنودی کے لیے ماہوار مقدار کی ادائیگی کا مشورہ مان لیں

عمر چور اور اغوا کار پراچہ کو ریشمی باڑھ سے باندھ کر ٹھڈے لگاتا موضع کے باہر مقبوضہ تھانہ تک لایا

زیر غور تجویز میں شامل معید عرف موچی کا رخ مری کے گاؤں ملیہ پور کی مسجد کی طرف تھا

مرثیے میں پتھری اور سانسوں کے ساتھ میتھی اور ورزش کا ذکر نہیں جچتا

ٹیچر کی آنکھوں میں آنسو دیکھ کر تحصیل ناظم پگھل گیا اور بڑھاپے میں اسے تھام لیا

چینی شخصیت نے لیدر کی منڈیوں پر تنقید کی اور فوری امور ٹھکانے لگانے کی بات کی

میں نوروز اور چہل سے پہلے ہما کے گھر چلوں یا روضے کی سیر کے بعد پتھر چوم کر

ساریو نے تنوع اور ڈھنگ سے لکھی غزلیں تاخیر سے شہر بھر میں بوکھلاہٹ کے ساتھ چھپوائیں

دواؤں اور اثاثوں کو سنوارا اور تیسر سیکھی اٹھتر درہم اٹھائے اور چوتھے روز چلا گیا

بھینس کھولو بھنوؤں کو اٹھاؤ اور گھڑیاں دیکھنے اور کڑھنے سے پہلے نشیب میں چلے جاؤ

بحثیں ماضی کا مدفن ہیں گستاخی بدروح اور جان لیوا ساز ہے

آئیے اسلحہ سیلز کے دوررس خسارے سے بچنے کے لیے شیشے کا بفرزون لگائیں

بھئی کیفے میں ادھم مچانے اور پمفلٹ کو گھورنے سے مکھن کی معیاد کے خدشے ویسے ہی رہیں گے

بےہودہ تیزی کے فشار سے قفس میں جھاگ بہائیں اور مہلک ذہن کی پیروی میں تامل کریں
اجڑے کھوہ کی کوکھ سے زوال جنم لیتا ہے اور ثریا پہ خورشید کا مکھڑا بے نقاب ہونے سے زندگی جنم لیتی ہے
مطہرہ دلہن نے ہمجولیوں کو دیکھا تو گویا خوشی ڈیڑھ یا دوگنا ہو گئی
اسلامی سال کا وعدہ پنشنر کو لڑکھڑا کر شل کر دے گا اور ایک آدھ بدسلوکی بھڑک کر وائرل زنجیر بن جائے گی
جاگنگ اور اتھلیٹکس اطلاعات کو براؤزر میں لا کھڑا کریں اور اور آئیز گوشوں کو بھولنا شروع کریں
نارتھ ضلع کے جھنگوی خیل اڈوں پر بھورے پھل کے نیچے بانجھ لڑکا سپردخاک کیا گیا
چھیلنے اور پیسنے کے تھوڑی دیر بعد بیڑیاں چھڑا کر ننھی چڑیا کے پنکھے سمیت بیلوں کی طرف جست لگائی
ے کے نرخوں سے چڑ کر مشرف نے کئی ڈالر بخشش میں دیے اور خیر خواہ پر بھڑک اٹھا
صوفی معشوق نگران کو نیچا دکھا کر ابھرے اور دیکھا دیکھی پلنٹی کے طور پر نور لوشن بھجوا رہے ہیں
کلیرنگ کے لیے عدلیہ کے سامنے وگ لگائے تو تیری بدچلنی لینڈ سکیپ کی طرح دھل جائے گی
لینڈ فلڈ نے مونث مچھلی کو بحیرہ عرب میں مہاجر بنا دیا اب راجن اور کیریو کا جال بچھے گا
کبوتر لیزر پڑنے سے روٹھ کر اڑیں اور انہونی سے بچھڑے بھی ساتھ آ جائیں
ڈانس اور بھنگڑا ڈالنے کے لیے بن سنور کر کسے کیفی اور عاشور کی طرح غصہ آتا ہے
شہر کی بسوں میں اوورلوڈنگ کا طرحی موضوع اور کچہری میں اس پر سزائیں ملنے کا احوال نیا نہیں ہے
چڈا صاحب لوٹ کھسوٹ کر سرمایے کھونے کو ہی موئسچرائزنگ مینیجمنٹ سمجھتے ہیں
عرس پر میں نقشبندی کوچہ میں گھی پہنچاؤں تو خدانخواستہ کوئی ٹریجڈی نہ ہو جائے
مینوئل پروگراموں سپیل چیک کا فولڈر ڈیلی غائب جدا یا مدغم ہو جاتا ہے
ینگ بوائے چھری کنگھی اور ترچھا ہینگر چھاؤں میں پھینک کر لاریب شوروم کی طرف چلا گیا
سکول کے نقشے میں بوسن میموریل سینیٹ کو ییلو پتھریاں گھیرے ہوے ہیں
خوشبوؤں والی جھاڑیوں کے افق میں جاری یوگا دوڑ اور کبڈی میں حائل ہے
بارش میں کوئلے کا تہتر فیصد فقدان ادھیڑ عمر غلام رسول کے لیے بےعزتی ہے
نوزل کی صفائی کے بعد بستر پر سوئے خان کو کیا سوجھی کہ ندا کے شوہر کو چھیڑ دیا
ٹھہرو تم جھد اور مہر سے ان غزلوں کو لکھ دو اپنی فیسیں لو پٹھان پڑھے یا پڑھوا لے
بھو کی فغاں پر چڑوں کا ماتھا ٹھنکا اور بھون کر کھانے سے پہلے دلنشیں مرغابیوں کے دریاؤں کے رخوں میں نکل گئے
سوچ بچار کے بعد ٹیوب کو رگڑیں تو راؤ کا نشانہ شاہ کے گلوز تک ہی رہے گا
تہوار کے نوخیز حصوں کے معانی کا اظہار پٹھے اور بازوؤں کے مقابلوں پر ہے

باڈی فٹنیس کے لیے چھابڑی خوان کی اوکاڑوی بھنڈی اور پھلیاں تناول کریں
حزب آفرین کے لومیرج کرنے والے سینئر ایڈہاک ملازم دین چاچڑ اس عرصے میں گاؤں پہنے آ پہنچتے ہیں
چاڈ میں غدار گینگز کا ظلم کئی گنا ابھار کے ساتھ آشرم میں قائم پیچوٹری میڈیکل سینٹر کے وقوعہ تک آ گیا ہے
خلیفہ نے سقوط منش آباد پر غوروخوض کے بعد ننھے بالغان کی چوگنی صفیں دیکھیں
مرغزار کلاں میں کچرے کے ڈھیر پرنیا کچرا ڈال دینے سے شہر کے بیچوں بیچ مکھیاں پیدا ہونگی
مہندی کا ڈھیلا اوس سے کوسوں دور گھسیٹ کر گھرے یقین کی اطلاعات کے ساتھ خسرو کی آغوش میں رکھا
پگڑی اور گدے کو دھوئے بغیر لیلی اچک کر کروڈ میئر کی بگھی میں جا بیٹھی
آن لائن ٹریڈنگ میں وم چیونگم سے دوشیزہ کو چھینکیں آنے میں کوئی مضائقہ نہیں ہے
روزوں میں ڈرائیونگ سے درخور تو کیا رکھوں عید پر جوڑیاں قدآدم پائیدار ابھری پگڑیاں سامنے ہیں
گیس کے سوویں نقرس کی فیس اولین تنفس کے جھمکے کے نیوکلیئس سے زیادہ ہے
بلی کی میاؤں سے بے سدھ چیل کا اکھڑنا اور لعاب سوکھ جانا دہرانے کے قابل نہیں ہے
جھائیاں اور چھائیاں ہونے کی وجہ سے رضا لتھڑے ہوے سبجیکٹ میں داخلوں کا دم نہیں بھرتا
فیری گھڑیوں کی نعمت سے واقفیت کی پرسش کے بعد بھڑوں کی فلم کا خول ابھرا
حروں نے شہابیے کو ساگر میں پھینکا اور پیداوار کے رزق کی ڈیل کرنے کے بعد گاؤں سے اوجھل ہو گئے
فجر کے بعد ورزش استغفار کی مقدار باوجود ضعف کے بے ناغہ علاوہ بروز جمعہ یکساں ہے
دبیز جھریوں سے کھویا انگوٹھا گزری مشق کے سہارے کیا دوسروں کے لیے سبزی بنا کر رکھے گا
ویمنز میں چڑچڑے ڈھونگ چلنے اور کھلنے لگے تو جوئیہ لنگر سموئے کلی کے گچھے لیے آیا
عبدالحئی میٹھا اور مٹھائیاں مانگنے کے لیے برتن مانجھے بغیر بچھانے میں کوشاں ہے
میں موڈم لے کر مون کے پاس جاؤں تو دریں اثنا دوسرا تھائی پروفیشنل چھین لے گا
دوم یہ کہ غوث کی مووی میں تھنک ٹینک کی فشنگ پر نیگیٹو رپورٹ پڑھوں تو رول بیک کروں
کچھوا اور گوہ دکھنے پر ڈائنو سارز نے جبڑوں کو ڈاؤن کیا اور مووایبل دیو کی طرح چڑھاؤ شروع کیا
داخلوں میں پیتھالوجی اور انگلش سے آگاہی کو ایک زود کوش اعزاز سمجھا گیا
فاجر دنیاوی خداؤں نے کفر اوڑھا اور پیغمبر کی بانگ کو لطیفے کا رنگ دیا
براوو کی لولی سوئنگ اور ملزوم پچیں گئی بھاڑ میں بھجن شالا کروائیں اور ایڈووکیٹ بھولا سے سنگسار کا پوچھیں
دوجے وبائی مرض میں ہڈی کی کوڑیاں جوڑیں چونکہ ریزروز میں آئیوڈین ہے ناں
جونا گڑھ کی سفیدے کی کرشنگ کی بو بھولیں تو فعلن کا اسقاط اوکھا نہ لگے
معید نے ساغر و چنگ کو مولا کی نعمتوں گنا اور لاکھا کی چھایا اوڑھے دیش کے اڈے پر وزن کا پوچھا

مشعر نے گھور کر جج کو دیکھا گڈا کھبے ماؤف بریتھلائزر میں چاقو سے پرویا اور سوچا کب تین بجیں اور میں چھم سے ویمپ میں آ گروں
گولڈ کی سرنجیں اور ویلوٹ کی کچھی پر تھو ہے کہ مبلغ چھ ماشے میں عاشق تغلق کے خرچ کا محور بنا
چیل اس فعل پر مصر ہے کہ پرتھوی بطخوں کو چھیل کر نوچ لوں اور پھاڑ کھاؤں اور ایں ویں ای لے اڑوں
لیبر کو سویاں ملیں اور اینگرو ہیئر ماؤنٹ میں ایشز کی سٹرلنگ آبدوزیں ایکسیڈنٹ میں اجڑی ایئر قوس میں نظر آیئں
میری آنکھو اور دانشو پروسیسر سے قم اور قل کی آواز سنو اور آؤ گگن تلے گھنگرو نوازوں کے مواد میں گھلیں
بلڈوزر سے گوڈی پر تھینکس کہا اور خویش سہاگہ پھیرنے کے بعد اس موسم میں تھوہر بوئے
فخر سے حافظے میں مغفرت کی شفاعت کے لیے اور عصیاں سے پاک منجھی ہوئی زندگی بسر کریں
سوئی ہوئی زلیخا سہیلی کی منگنی کی خبر سے اٹھی اور بروز بدھ نئی چوڑیاں باڑہ سے خاصی رقم میں خریدیں
بش کو گوگل کا ویب پتہ دوں تو خارجہ پالیسی میں پوچھی گئی بھیڑیے کے خلیے میں سونگھنے کی قیود سے جڑی کہانی چھڑ جائے گی
عملی دلچسپی کے باوجود بونس صفر بنیں تو کیا گوندھیں اور کیا نوش کریں
فارورڈ فلیچر میں بلبل کو کھلے آسمان کے گشت میں بم یا ایسڈ سے داغنے کی کوشش کریں
بچے گود اٹھا کے کوڑھ اور قحط سے بچاؤ کے لیے دعاگو ہیں آپ تنسیخ پر بدھا کا پیچھا چھوڑ دیں
خادم ابرار کالج کے انٹر سیکشن کے نیچی بلڈنگ میں بائیولوجیکل ٹیکنالوجیز کے اصلاحات کیلئے انظباتی قوانین بنا رہا ہے
آٹھویں ٹورازم ڈیویلپمنٹ کنونشن کی کمپیرنگ ڈائرکٹر زیدی جیلانی نے کی
صوبائی وزیر خزانہ رؤف نقوی اور آرگنائزنگ کمیٹی کے لیجنڈری صدر جاوید ابڑو نے زعما اور معززین کو بتایا کہ درسگاہوں میں لیب کی ضروریات زیادہ عذاب بنے ہوئے ہیں
لال پھولدار جوڑے میں نیم عریاں اجسام منحوس ڈگر کی جانب پیہم رواں ہیں
ملازمین کی خوشیوں کا رازداں آسان پیشگوئیاں کرتا ہے
سوکھا نشاستہ جوس میں ڈبو دو
مسلمان کھلنڈرا وزیر اعظم موروثی اتھارٹی رکھتا ہے
ضیغم چوہان پیشگی مداخلت کے بغیر مفروضات کے تعاقب میں الجھا رہا
اشفاق چنیوٹی امریکی ایوارڈ لینے ہیڈکوارٹر گیا
مجرم کو ڈیتھ اسکواڈ سے اڑانے کی سزا دی گئی
ماں نے بینگن خرید کر پکائے جو جل نہیں گئے

ہنسیں گائیں ہوش میں آئیں رونے کا یہ وقت نہیں ہے

اوزار سے چھلکا نہیں گیا چنانچہ چبا کر بیج کھایا گیا

سونے کے صاف ڈھیلے نصف میل سے لایا

اس نے اونی سوئٹرز دھو کر دھوپ میں رکھنے کے بعد بتایا

متوفیہ کے نوحے کے بارے میں کیا بتاؤں

چین نے شیڈ راؤنڈر کو برابر امداد دینے کا تہیہ کیا ہوا ہے

اللہ تعالی سے خوف استغنا کا تسلسل شو کرتا ہے

فقیر نے حجت کے بغیر جان جوکھوں میں ڈال کر آدمی کا تعاقب کیا

بڑی بحث اور تجزیہ کے بعد فیصلہ ہوا

ماخذ کا تعین نابغہ شخص کی اچیومنٹ ہے

جیز کے بغیر زندگی کیسی دکھ کے ساتھ گذر جائے گی

وزیرداخلہ قیصر چنگیزی اسمبلی کے اکیسویں اجلاس میں مصروف ہے

جس دھڑے سے بنولہ روغن کیلئے محلوں میں فروخت ہوا یہ انداز غلط ہے

چند دن شیو نہیں کیا سونگ مشین کے کام میں مصروف تھا

روزبروز برج کھیلنے اور کاغذات باندھنے سے رنجش کچھ کم ہو گئی

جلسوں سے انقلاب آنے کا سوال نہیں پیدا ہوتا

کوٹھی کی اداکاراؤں پر مبنی فارسی مضمون شریف اور کمزور عورت کیلئے غیرضروری ہے

شیریں دیومالائی داستان کا بآسانی دل میں جاگزیں ہونے کا خدشہ ہے

عام چھوٹے ریسٹورنٹ کی مشروبات اُٹھانے سے غذا رغبت سے کھائیں

ٹوئنٹی ٹوئنٹی ٹورنامنٹ کی وجہ سے ہڑتال کا اعلان پانچویں مرتبہ رہ گیا

کاشف قزلباش تشنج کے سبب معالج کیلئے بہترین معالج کے پاس گیا

موسم گرما میں قندیل کے شعلوں اور بجلی کی روشنیوں سے گلیوں اور گھروں کو روشن رکھو

لیفٹننٹ سعید جعفری جھنجھلاہٹ سے مغلوب ہوا تو بلڈ پریشر کا نسخہ لیا

سپرنٹنڈنٹ ڈیوڈ سمتھ انسٹھویں چیمپئینز گیمز کا پلیئر ہے

تحقیقات کے درمیان قاضی سچل سومرو کی پرائیویسی توجہ سے دیکھی گئی تو تمام تحقیق کرنے والے دنگ رہ گئے

مچھیروں کے آزاد گروہ نے پیشقدمی کرتے ہوئے سیکورٹی کاروائیوں کے برخلاف جائیداد واگذار کرنے کا ڈیمانڈ کیا

جوجوبا کی دھونی کرنے والے باقی پیروکاروں کو سمجھائیں کہ یوں بےمعنی خوشحالی تلاش کرنا چھوڑ دو

اسلم بودلہ کے ترجمان نے کہا کہ سول سوسائٹی اپنے مشروط حق لینے میں سرخرو ہوئی ہے اب چودہ عمرقید بولروں کی بولنگ کیلئے پریذیڈنٹ کو متوجہ کرائے

کونین چمچ سے نکالیں اور مرغن شلجم سائیڈ پر رکھیں

رواں پنج سالہ بجٹ میں بچیوں کے استفادے کیلئے جدید رہائشی گرلز مڈل کلاسز منظور

دارالحکومت میں کھاد کے مینوفیکچررز نہ روٹھیں اپنی ترجیحات واضح کریں

پڑوسی سرزمین میں بوریت کی مشقت برداشت کرنے کے بعد بالاخر بینظیر بھٹو سمجھدار ہو گئی

نومنتخب کوآرڈینیٹر نے چار علاقائی تاجروں اور زمینداروں کے مناسب تفریحی سہولت کا خاص احساس کیا

پڑھائی خاموش مطالعے اور تعوذ و پنجگانہ سجدہ کا طریقہ باشعور ملاؤں سے سیکھا

مدرسوں میں ترمذی شریف عموماً آخری درجوں میں پڑھائی جاتی ہے

چچا چچا ذاد بھائی اور بہن مجھ سے فرینڈلی ہیں

جوائنٹ ڈیویلپر مبشر عظمی نے خاموشی سے اپنے فرائض سے استعفے دے دیا

بالغوں کی مونچھیں نمودار ہونے لگیں تو ان میں ہیجان اور ضد عروج پر ہوتی ہیں

ایسی حویلی مغلیہ بادشاہ کی ہر بیگم کو مہیا کی جاتی

ہوم ٹیوشن سے بچے کے گریڈ اچھے ہوئے

عراقی عوام حصول عدل کیلئے دوبارہ اتحاد کرکے لڑیں

راستے میں ایسی گھنگھور گھٹا چھائی کہ ایک لمحے کیلئے میں قدم اُٹھاؤں تو نہ اٹھے

جرگہ چاہے تو مجرم کو عدالت کی سیڑھیوں میں پیش کرے

لڑائی پر مصر چئرمین معین سیٹھی نے مجھے لکھا کہ ہم تمہارے رویوں سے متنفر ہو چکے ہیں

شنید ہے کہ بوبی پنچولی کا حسن نزلے کے سبب مرجھا جانے کو ہے

کچھ حضرات گریجویشن کے لیے ممد و معاون کانفرنسوں میں اولولعزم ہو کر شعوری شرکت کرتے ہیں

بیوہ نسرین سبزواری بے اعتبار اور بے ہودہ مردوں کی میٹھی خوشخبری پر دھیرے سے ڈھیر ہو گئی

فرانسیسی نژاد رمزی عثمانی نو دیگر بیوروکریٹس کے ساتھ ایفروایشین سمگلروں کے تعاقب میں ہے

برڈ فلو سے متاثرہ گھریلو مرغا اور مرغی زیادہ مقوی نہیں ہوتے

حریفوں جیسے ممالک کا وجود لخت دولخت تقسیم کے عذاب سے متغیر ہو سکتا ہے

گہماگہمی کے موقعوں پر حیران تصویر نہیں ہونا چاہئے تماشائیوں سے دل کھول کر مزے لینا چاہئے

محسنہ صدیقی موثر ڈرامہ لکھے بغیر غیریقینی مخدوش حالات کی وسعتوں میں گم ہو گئیں

متنازعہ نظریے کی پیچیدگیوں پر چوہدری ساجد رزاق نے فرمایا مفہوم شرحوں سے واضح ہو گا

گراؤنڈ میں سیمینار کے انعقاد کے دوران ناظم ساجد راہو دھڑام سے نیچے آکر بیہوش ہو گیا

سیاسی بیداری روزنامہ جنگ کی پہلی ترجیح ہے جو ہوشیاری شمار ہوتی ہے

سٹوڈنٹ کیلئے ڈرائینگ اور جیومیٹری پر دھیان دینا بقیہ اشخاص سے اہم ہے

مہنگائی الاؤنس دیانتدار فاریسٹ افسران کا استحقاق ہے

ریزرو شماریات معاشی بڑھوتری کے رجحان کی ترجمانی کرتی ہیں

ماحول کے رونق کی بدمزگی میں تبدیلی صحتمند سوچ نہیں

ادھار کیلئے بغیر ہچکچاہٹ کے الحاج اعظم یزدانی سے ٹیلیفون پر پوچھیں

پہلے پیریڈ میں ضعیف سرفراز دہلوی کو مہلت دینا نا گزیر تھا

مثلاً آٹھ یوم تک اصحاب کہف کا خشوع و خضوع واقعی مومن کا محیرالعقول ایقان ہے

تفتیشی ایجنسی نے یہ منظر دیکھا کہ پولیس نوجوان لاٹھیاں برساتا مشتعل جلوس کے پاس پہنچا

پروگریسیو کارپوریشن کا متحرک سیلزمین ایچ ایف خلجی کی کامیابیوں کا ریکارڈ معقول ہے

بریگیڈیر اکرم کولسری نے سوشل سروے پروجیکٹ کی ملازمت تدریس پر تیاگ دی

رینجرز کے سپاہیوں کے اسکواڈ نے کلاشنکوفیں نواب کے اداروں کو دیئے

دور افتادہ علاقے کا شاعر رشید بھوجپوری آٹھ مہینے بعد بڑی کوششوں سے دریغ نہ کرتے ہوئے شائیں سے اپنے عزیزواقارب کے پاس پہنچا

آؤ سانحوں سے بھری ہوئی بدحالی کی چادر نوچیں اور انحطاط کے افراط کی مخالفت کریں

ارجمند اقبال اور آسیہ جبیں صورتحال کی بہتری کیلئے پانچ ہفتے منتظر تھیں

بسا اوقات بشیر اپنی ہمشیرہ اور اہلیہ کے ساتھ یہاں انوکھے ملبوسات کے خریدوفروخت میں الجھے رہتے

مراد خاں تیزرفتار ٹرین کے دریچے میں مایوس بیٹھا گذشتہ شب کے واقعہ پر حیران تھا

اخلاقی امیج بنانے کیلئے اور جھگڑے سے بچنے کیلئے رضوان اپنے پیشرو خان ذادہ کی کوٹھیوں کے دورے پر گیا

غیرملکی آقاؤں نے مقروض قومی وزیر کا معاوضہ ڈکٹیشن بھیج کر آدھا کروڑ کر دیا

سود کی تجارت کا سودا مودبانہ مانگ لینا نوزائیدہ مملکت کیلئے مثبت ایفکٹ نہیں لایا

آؤٹ آف وے ٹرانسفر کی لاگت لینڈ سلائیڈنگ کی مرہون ہو گی

میجر خلیل یوروپیئن ایئر ہوسٹس کی زلفوں کے سنگھار اور تزئین میں گرفتار ہو گیا

ٹشو کاغذ سے چہرے اور ہاتھوں کو پونچھنے کے بعد لارا سیدھے گراموفون کے پاس بیٹھ گئی

تیز رفتار سونامی سے بھربھری چیزیں سیراب کر لوں

آرتھوپیڈک ریسرچ کے علاوہ فنگس میڈیسن پر بھی بہت سوں نے خوشگوار تجربات کیے

ہنگری کے بزدل سکواڈرن دیول نے لوہے کی دس کمپنیوں میں چیتھڑے ٹھونس دیئے

حضور نبی آخرالزمان نے اللہ کی مشیت سے فریقین کے قضیے زبردست فراخدلی سے سنے اور احسن انداز سے اقدامات کئے

غنا کا دولت سے ایسا تعلق نہیں اپنے اہلخانہ اور ساتھیوں کے ساتھ قناعت ضروری ہے

بروکلی شیرٹن گروپ نے رقص میلہ چھوڑ کر ایلوویرا وومن فلیش کنسورشیم بنایا

پرتعیش مغلوں کے با عمل قوت کی ریگولر فوجوں کے سربراہ نے الزام منسوخ کر دیا

پنکھا سنبھال کر مسز عشرت چست میرون لباس میں سیرگاہ کی جھیل کا نقشہ بظاہر تخیل سے بغور

بھانپ رہی تھی

بھانجے سے کہنا شوق سے پرفارم کرنا اور کہنا کہ انسولین کا نفوذ سوجھ بوجھ کے ساتھ توڑیں

برفباری میں ڈھابے کے چولہے پر رکھی چھوٹے اور گوبھی کی معمولی کھچڑی ناقص قسم کی ہے

جنوبی کاؤنٹی کا بھارتی کوچوان موہن زلفی پرائیویٹ ہومیوپیتھی اسسٹنٹ جیم ریگن کی پڑیاں دبائے بیٹھا ہے

باریش اہلسنت حفاظ ظہیر باجوہ اور راشد چغتائی نے شعرا سے نظم کی تصحیح کرائی

کنڑ کے گلریز خان کے ٹریلر پر مخصوص چمڑا اور لوہا دھرا ہے

درد اور خوفوں سے ششدر غریب اور لاچار بوسنیائی مسلمان مفاہمت سے بازیاب ہوئے

کمسن ملائیشن وفاقی وزیرخارجہ داؤد احمد توصیف سیاسی اور اجتماعی چیلنجز سے بتدریج اور کماحقہ ایثار کے ساتھ پیش آتا ہے

ساتھیو انتھک محنت کا احترام بے گھر اور مجبور مخلوق سے سیکھو

نہایت باادب ہو کر مشائخ عظام سے انسانیت کی شناخت کا مسئلہ دریافت کرو

بے لوث اور مستعد ساتھی بوقت ضرورت بانہیں کھولے وہاں انہی مہمان کے ساتھ موجود تھا

انوکھے اور بیباک رفیق کے اداس ہونے سے واقعی دل کو دھچکا لگا

ساڑھے چوبیس ہزار کی غیر معیاری جیولری کے چکاچوند سے بیوی کے رویے میں باآسانی تبدیلی لائی جا سکتی ہے

ٹوائلٹ اور سیورج کا اخراج بگاڑنے سے الجھاؤ کو مزید براں بڑھاوا ملا

سنئے تمہیں خصوصا معلوم ہونا چاہئے کہ وحشیانہ جارحیت نے اشد کنفیوژن پیدا کیا ہے

ایئر فورس کے یرغمال اہلکار کو بلا جھجھک ہمیں پیش کیا جائے

گلگشت کالونی میں ایڈورٹائزنگ ایجنسیز کے ریسپشنسٹ نے کہا میوزک سنئے

بڑا زوردار دھماکہ ہوا اور آرٹلری میدان کے بیچ میں گھرا سوراخ پڑا

ساٹھویں سعودی بیڑے کے بوئنگ چارجز میں تخفیف زیرغور ہے

میرا مرزا فاروق امجد سے دیرینہ تعارف ہے

سوزوکی کے مختلف حادثے بڑے پرہجوم سڑک کے مقام پر ہوئے

انفرادی آڑھتی لاکھوں کا بیوپار حصص کمیشن کی قدر کی صورت میں کرتے ہیں

رہائش گاہ کے ساتھ واقع باغیچہ میں مزیدار دھیمی خوشبوئیں بکھری ہوئی تھیں

پینسٹھ برسوں کا معطل شدہ میچور قانونگو شفیق باقر گومگو حالات سے دوچار ہے

مترجم وہاب جہانگیر ملازمت سے برطرفی کے بعد پچھلے پیر کو میڈیا دفتر میں عود کر آیا

قاسم سٹیڈیم میں جلسہ جلوسوں کے سلسلے ممنوع ہیں البتہ چھبیس کلومیٹر پر پوجا کی جا سکتی ہے

اس سے قبل بیگ امروہی کے ساتھ پالمر ریمبو کا لہجہ دوچند سخت تھا

پارلیمانی تاریخوں میں مدہوش پنجابی وزراء ہمیشہ رخنہ ڈالتے رہے

انگریزوں نے فلسطینی نقشوں سے جنگلوں گلیشئیر چیڑھ کے درخت اور سیروسیاحت کی مشہور جگہوں کی نشاندہی کی
ٹھیکیدار نے مزدوروں کے مطالبہ پر مارچ کے پروڈکٹیو منافعے منگوا کر آٹھوں لوہار فارغ کر دیئے
فارم مزارعین باقی اجناس کے علاوہ سبزیوں میں گاجر سرخ مرچ لیموں اور گندم باجرہ کے نرخنامے آویزاں کریں
گیہوں اور سبزیوں کے سوکھنے اور سڑنے نے لائیوسٹاک اور زراعت کو نقصان پہنچایا
بیچارے مجذوب کو قیلولہ کے بعد خوراک کی جستجو ہوئی
دکھی محبوبہ مسز اوڈھو کے آنسوؤں کو دیکھوں تو آنکھیں سوگوار ہوتی ہیں
پر اُمید ہوں کہ میراں بیگم کا گرانقدر وظیفہ ادھورا نہیں رہے گا
مگر سکھوں نے آباواجداد کی جھیلوں اور راجباہوں پر حملہ کر دیا
پولیو کے انجکشن کی قیمت اورنگ کی فہرست میں نہیں
چونتیس سے اڑتالیس گینگسٹر چوہوں کو معدے کی بے چینی سے قے ہو گئی
انکشاف ہوا کہ شیوخ کی بیوگان معذوریوں کے باوجود کبھی زیردست نہیں رہیں
میری نظر میں جنگی قیدیوں کے پرخچے اڑانے کا اعتراف کرنے کے کچھ بھی فوائد نہیں ہو سکتے ہیں
پوٹھوہاری تھیسس کا وقوف مورخہ اٹھارہ جون سے معیوب ہوا
بڈھا نانگا عرصہ سے اندھیرے پنجروں کے جکڑن میں داخل بحران ہے
بے وقوف گونگے کی معیت میں متنوع گارلینڈ کی حریت آڑے آئے گی
پھاہے بنانے کیلئے السی اور گوبر کی وافر مقدار کوٹھے پر پہنچا آؤ
گوہر فروش نے مسخرے سے اریگیشن شیلڈیں اونے پونے میں خرید لیں
لادینیت کے تحریروں کا خوگر مبہم ڈپلومیسی سے جھگڑالو ہو گیا
گدھوں کی انڈسٹری ہارویسٹر کے قبضے میں مقید ہے
ممبر رزاق میلسوی کی روش سفید سینڈوچ کے تحفے بھیج کر بدلو
قوی بہبود کی آرزوئیں گڑگڑا کر منفعت سے تسخیر کریں
مخدوم شجاعت لدھیانوی کا استفہامیہ لفظ جنگجویانہ ہے
غوں غاں کرتی ہوئی رینگتی بچی نے جھولے میں پیشاب کیا
فیضان تخلص رکھنے والا شاعر زیرآب گنگا نہر پر بگڑ گیا
رہائشی ورچوئل گروہوں کی وباؤں پر صدیاں ضائع ہوئیں
گیگا روور اینڈ کمپنی نے سلیولائیٹ کلوننگ کے پراسیس سے اپنے پروفائل کی نمو کی
وحشی تھانیدار ٹیڑھے مذاق سے زور سے قہقہے لگا کر ضلعی گودام کی حفاظت کرنے لگا
اومیگا والرس اور کچھوؤں کی خودکشیوں کی نحوست ماحولیات کے چیلنجوں میں سے ہے
کوسموس کے دراوڑی ذخائر نقش چھوڑنے سے اجاگر ہوئے

منشی خاقان ہاشوانی کی نئی انگوٹھی کے کروفر سے عزت کا تاثر ٹھیک ہوا
شوخ مچھلیاں اور نحیف مرغیاں عیاشی کے زمرے میں آتی ہیں
سیٹھ ابصار دوعالم کو اجرت کے اضافے کی توفیق نہ ہوئی
بالغ منشیوں کے سروں میں کھجلی رسولی کا شاخسانہ ہے
دودھیل امرتسری گائیوں کے نشوونما کیلئے جوار بیجوں سمیت بکھیر دیں
بقایا حدیث کے تراجم صفحہ چونسٹھ پر دیکھئے
سالہاسال سے لوڈشیڈنگ کا ازالہ دعاؤں سے کرنے کا مژدہ سنایا جاتا ہے
ہتھیلی پر پاؤڈر کی آمیزش سے افیم کی موثر خوشبو آئی
گوجری بھاشا میں شعری قانون کے لٹریچر کو روزافزوں ترقی دی گئی
بقول شخصے جمہوریت کی حیثیت حوض کے ساتھ خودرو گھاس جتنی ہے
جب مصوروں کا صدیوں کا اثرورسوخ نکھرا تو متعدد ماہرین کو بوجوہ حیرت ہوئی
مضبوط اعتماد سے افیون کے مطلوبہ استحصالی وفود کو خوفزدہ کیا جا سکتا ہے
واقعتاً لاہوری طنزومزاح کی بوچھاڑ سے گناہوں میں اضافہ اور طبیعت میں اندھادھند رچاؤ آ سکتا ہے
رانجھا لقب پانے والا آبرو مسخ کرکے ازخود سیڑھیاں موڑ گیا
سنگین جھگڑوں کے کیسوں سے بچنے والو کھڑے ہو کر آنکھ جھکاؤ
ترچھی فگار یادیں لہولہان یہودی کو ہجران میں کھچ کر لائے
رحیم میرٹھی کو چاہیے کہ وہ مرشد کو موسیقی کے تقاضوں سے روشناس کرے
فلسفے پر ڈیبیٹ کیلئے ریلیکس ہو کر پڑھئے
نخوت کی ترغیب انسان کو اندھیرے کے منجدھار میں ملوث کرتی ہے
مصنف کے گھنگریالے گیسو ان کی عارضی مردانگی کو ڈھانپ لیتے ہیں
جارج شیفرڈ ستانویں میراتھن کیلئے جی جان سے حوصلے میں ہیں
جوڈیشل ایگزامینر منتخب بچھڑوں کی پراسسنگ کی زور شور سے ممانعت کرتا ہے
صارفین اپنے کمروں کی واجب بلوں کی بہتری کیلئے مشین نوزلوں کو سنبھال لیں
افسردہ لیڈی عرفان گاڑھا پراٹھا تمام ساتھیوں کیلئے ڈھونڈ لائی
میران بگٹی شائد امرتا سنگھ کا بھیس بدل کر منڈھے پر چڑھ آیا
سابق بیعانہ اور ڈیسک پرچیز کیلئے گڈ گورنینس اپنائیں
بیلدار اپنی کمینگی حاوی کرنے کیلئے عارضی طور پر نزدیک چورنگی سے کھسک گیا
دکھوں اور نیرنگ انتہاؤں کا دارومدار پرخلوص پالیسیوں پر ہے
مافیا ایجنسی شیلٹر بلڈوزروں کی سمگلنگ پہچانے بغیر واویلا کر رہی ہے
وومن کالجوں کی سلور جوبلی سرگرمیوں کا صوبوں کی دوغلی سماجی موضوعات سے خصوصی ٹرانزیکشن ہے

راقم فیڈرل فوریکس جوڈیشری کی تضحیک کے خاتمے کیلئے ویلفئر کنٹریکٹ کا خواہشمند ہے
سٹوڈیوز میں فروزاں شیل کے پتھروں کا گھناؤنا ارتعاش کچلے جانے کو ہے
غیر جانبدار افغان بھابی کے ساتھ ننھی گڑیا کا موڈ بدلیں تو خفت رفو ہو سکتی ہے
سندھی ایتھلیٹ ثاقب ابڑو اچھل کود دوڑنے اور بیٹھک کا درس دیتا ہے
موہوم پیشنگوئی دیوار اور لوحوں پر مرتب کرکے لکھو کہ ہیروئین کی انوکھی عفت زیور تک موخر ہے
ریچھوں پر مواصلاتی اپروچ معجزات کر دکھایا
تصور فرمائیں کہ لاکھ دلسوزی کا سکنجبین پیئں لہو کا خمار آئے تو محبوب بھی لاٹھی دکھا جاتا ہے
سموسے اور مرچیں کھا کر رابعہ کے گلے کے عضلات اور نسیں پھیلیں
سادھو نے الہیات کے ابلاغ کیلئے فقر کو اپنا اوڑھنا بچھونا بنایا
میرزا شہاب دیوبندی نے نویں بیچلر فالوآن کی تہیں کلیئر کیں
لاڈلے چودھری مبین گردیزی کے بسیار لاڈ سے اولاد قبیح بداخلاق اور موذی امراض کی شکار ہو گئی
کنفیوژ پیروکار اڑسٹھ ایام سے کوٹھڑی کے تھڑے پر جھاڑو دی کر خفیہ محفل سجاتے تھے
سکھی نرگس صغری لغوی تحریر سے فریب کھا کر خود بخود مصیبت میں پھنس گئی
مدعیان جلد از جلد کھلاڑیوں کی بیویوں سے شوربہ اور قورمہ کا تناسب سیکھیں
سبسڈی مخفی رکھنے سے نجی فصلیں بھیگی ہو کر آشوب کی بھینٹ چڑھ گئیں
غیور یانگ نے جبری مصروفیات اور بجھائی ہوئی آگ کے دھوئیں سے بپھر کر بلغم اور خون اگلا
مخیر لیاقت چدھڑ نے کہا میں چل کر ننھیال کے ڈیرے تک ہولوں
شعور کے زیر اثر نیگیٹو تصورات پتھالوجسٹ کے تجزیے کیلئے ایک گتھی ہے
رفیعہ نے اقلیتوں کے مفاد میں جنرک الیکڑانکس ایشوز کے غصب کرنے کا اقرار گول کر دیا
بائیں بازو کے نامور بلوچی لیجنڈ فعال اور موثر ثقافت پر اپنا غیرنمایاں ایقان دہرایا
غذاؤں تجزیوں کے سازگار مطالعوں کا فن جغرافیائی اندھے پن سے دفن ہوا
زمین کی گزشتہ تیرگی ہولناک سیسمک طغیانی سے بے اثر ہو گئی
خواجہ کریم کو بولو لاابالی نونہال کی تولید چائلڈ ہوم میں ممکن ہوئی
منشیات کے نیو کنگڈم کا تاجدار رہنما اتھلیٹس کے فرسودہ سامان کے ہمراہ جلدازجلد منتقل ہو گیا
جھونپڑی کے مدھم اندھیروں میں عشاق کی فوجیں لذت گناہ سے رقصاں تھیں
مگسی قبیلے کی سست فقیر نیسچن کے حصے میں تقدس کا ترشول آیا
ٹیلیویژن کے دلفریب افسانے اور انشائیہ فقروں کے ہجو سے فلمساز بڑے محظوظ ہوئے
انشااللہ ہفتے عشرے میں تیرے ٹائفائیڈ کے جھوٹ کی حقیقت دکھائیں گے
مس رتھ ایلیٹ کی ازدواجی و روحانی ناامیدی کا انحصار گھمبیر بدشگونی پر ہے
نرسیں شیرخوار مریضوں کو جھولوں میں بٹھا کر تھوڑے شیمپو سے دھولیں گی
اینگزائٹی ڈزاسٹر ذیشان کی نیلگوں روحیں چڑچڑا کرتی ہے

ہجر کی تلخیوں کو بھولو اور فلالین زلفیں تولیے سے صاف ستھرا رکھو
پہلوٹھی دیوی کی مانوس لومڑی کی صحت آئیندہ لمحوں میں ثقل ہونے سے زچ ہو گی
نگہت طیبہ سنگل تھی اب نقاش کی وفاؤں کے ساتھ زوجیت کا تعلق ہے
گینز ورلڈ ریکارڈ ایڈڈ ہوئے آٹھواں جنریشن ہے
پیٹھ پر کھجوریں لیے گجراتی باشندوں نے جھگی کی ڈیوڑھی میں جائزہ لیا
استغاثہ میں نغمہ نے رحم کا تکلم ظاہر کرکے نافذالعمل کاروائی سے گریز کیا
داؤدی افواج کے ٹھاکر ونگ کے لشکر نے نوے میل فی گھنٹہ کی سیریز ایگلز کو تفویض کی
چرانے سے قبل دودھیا گدھی کی گداز پیٹھ پر گتھا پھیرتے رہو
مٹھن سرہندی نے فیڈریشن کے ساتھ الحاق کی وفاؤں کو ایفا کرنے میں جلدی نہیں کی
اس موقعے پر کینیڈین افراد شہروں سے ہنگامی انخلا بھگت کر خوار ہوئے
مگ میں وہی پھیکا محلول چوہیا کے غدودوں کو مجروح کرنے کیلئے تھا
ایشیائیوں کے زیرانتظام کانفرنس میں پچاس داڑھی والے تبلیغی بزرگوار مبلغین کی طرح بولے
لچک کے باعث فرعون جیسے گنہگار جرنیل کو آرتھو لگژری قونصلیٹ بخشی گئی
ماؤزے ژنگ جیگوار مشینری کی سروس چھیاسیویں ٹرانسمیشن کیا مضحک ہے
انگریزی فوج کے پیچھے خشخاش کا ہیوی ذخیرہ ہے
ریسلینگ شاہراہ پر سندھیوں کی تصویریں کھینچیں تو انہیں دیکھے بغیر اپوزیشن گرویدہ ہو گئی
تنولی ہاؤس کا بورڈر معاذ شبلی کا شاگرد ہے
وائرلیس ریونیو کی منسوخی سمفنی ڈائیگرام کو موصول ہو گئی ہے
محسن کو ہارڈویر رقوم کی پیشکشوں کی نوعیت پر شبہات ہوئے
بسیں ٹریفک کے نوگو ایریا میں بوم کر رہی ہیں
سیب اور امرود کے بیج مندوبین کے مسوڑوں کیلئے نسخے ہیں
شیعہ حمایت کی نوید جہاد کے فروغ اور ٹریژری تمغوں کیلئے ہے
وڑائچ رضوی کی بھاوج بولیں ذوالجناح کی سفیدی عشر کی سعادت ہے
مفکر کی فقہی توضیح جا معیت کے شبہے کی توسیع ہے
تیزگام پہنچنے کا شیڈول بعدازاں صدیقین سے حاصل کیا گیا
سوم درجے کی جعلسازی کا بھیڑیا تھیلی اُٹھایا اور اپنے پڑدادا کے جلو میں دب گیا ہے
گنگولی کو منشیانہ ٹیکسیشن ویجز کے برابر میسر ہے
چیچک کے معالجہ کی کوششیں ناکام ہوئیں تو رازق گوجر کا انتقال ہو گیا
روات کے آغا مشفق کوفی کے تغافل سے کھوکھلے انتخابات محض داغ بن گئے
آرکائیوز کے ہجوں کو بازیچہ بنانا غایت سماوی قصور نہیں
نا اہل لکڑہارے کی غفلت سے فردوس کا آشیانہ بچھڑ گیا

کلثوم نواز کا واحد مشن خاوند کی مصروفیت کیلئے اپنی صبحیں وقف کرنا ہے

موبائل ٹھگ فکروعمل کے وقفے کے دورانئے میں بیگج کی تجوریاں اکھاڑ لینے کے چیمپئن ہوتے ہیں

گرل فرینڈ کے ساتھ صبح گلشن کی سیر کا خیال ٹھکرا کر چیف کو قرار آیا

میگزین کے ابواب ادھیڑنے کیلئے مہوش کو رہنماؤں کی آشیرباد حاصل کرنے میں کوئی باگ نہیں

نوبت یہاں تک پہنچی کہ دھاگے کے ریشوں میں دھری لنگڑی گلہری اُدھر انگور کے پڑاؤ کو عبور کرنے لگی

دلربا کو آیوڈین کے نرخ اور ٹھیلے پر رکھے ہوئے اڑھائی کلو بنفشی آلوچہ کے بھاؤ ارینجڈ کرنے کی شدت سے آرزو ہے

شمس شوگر اور نزلے کی بھڑاس کے ردعمل پر اُٹھا تو تقدیر نے بے کسوں پر کرم نچھاور کیا

سدھیر نے نااہلی کے دعوؤں کی گڑبڑ پڑھی تو کہا اس تو اچھا ہے کہ اس کے نخرے اٹھائیں

سولجر شعبہ جرم کے پیشے میں ڈھول پیٹ کر دخل دیتا ہے

ندیم کے دماغ میں بلوغت سے ہی اجنبیوں کے تحفظ کی بڑی فضیلت ہے

اکثریت کی الفت کی زباں گمبھیر نہ بنائی جائے

پھوپھے کے بڈھے مرغ کو خشت کا ڈنگ چبھ جانے کی الجھن تو ہو گی

الیکشن میں مغل امیدوار مگرمچھ نے پستولیں نکالیں اور فیسوں کے جوڑ کا تصفیہ کیا

معاصر اہل تشیع کا ایوبی امن بیڑا نیچر کی طرف سے ودیعت احساں ہے

کھسے پہنے بھینگا فوسٹر نیزے کے دباؤ کو گڑگڑاہٹ کی گونج کے ساتھ جھیلے

تبدیلیوں کی لغزش سے نڈھال بھورا ہیجڑا شفقت و اخوّت کی لینگویج کا خواہاں ہے

قلم کو لاحق بغض اور جہل کو رشدی نے الوداع کہا

بدھو فین نے شیشوں کے کرایوں کے ساتھ سمجھے بغیر سمجھوتہ کیا

ہدایتکار شہزاد نے بلیچ اور موئسچرائزر کو بلڈوز کرتے ہوئے جوہی خاندان کی تجوری میں اوورہیڈ بڑھایا

تعجب ہے والڈ ہڈ نے سخن اور وژن کو فروگزاشت کرکے سیدھا تصوف کی معرفت کا حلقہ اختیار کیا

کھمبے پر پڑے سوکھے گچھوں میں کیڑوں کو ہویدا دیکھ کر فدا ہنسنے لگا

فاؤنٹین کوریج سے روشناس مونچھوں والے اکھڑ صابر نے المیوں کے بعد عجز اختیار کیا ہے

تہذیبی تشبیہات مدعا بیان کرنے کا جوہری نتیجہ ہے

چیمہ نے پھلوں کے گولوں کو گڑھوں میں سمویا اور کیڑے مکوڑوں پر پرزور شعاعوں کی تیاریاں کیں

مجرد رئیس ہاشمی نے اٹھ کر زوجین کے مابین میزیں نوڈلز اور چابی نوازنے میں پہل کی

کوچی گڑھ میں مینڈھا اور گدھے دیکھ کر ایک درجن عمر رسیدہ بھیڑیں پارلر میں سے گذریں

ڈینٹل گینگ کا ڈراپ منافع سوبرز کا مہنگا نشاں لے گیا

مفسر کے نسخوں میں گلسرین اور فاسفورس والا فوڈ غدود کے انفیکشن کیلئے مہمل شگون تھا

نیرو ایڈن سائبرپیڈیا کی آدھی قدریں لے کر بگ ریسیور کے مواقع کھو گیا

لیمپ بجھنے کو مرلی حیات دھند کی یلغار سمجھا

فرقہ خیرالعمل کے نبوت کے اعلامیے سے بڑوں کے تحفوں میں نکھارا گیا

لغت کی تکمیل نبھانے کیلئے بیٹھے ٹریبیونل میں فنون کی خوشیاں مفقود تھیں

شیلڈ ٹریڈ کے گوشوارہ کے اواخر میں فیلڈ ابزرور حنیف نے مرغوں کا ہیضہ گیج کرایا

قیم خضر نے نشے کے اشتعال میں آکر بیلچے مارے تو چھن سے خم بکھر گیا

سیریلز چیفس کولن ڈاگ بیڈ نے وقفہ کرکے فیلو ٹرفس بھیجے تو اس نے اسے مڑ کر چھالے دکھائے

حزیں سیدہ نے کہا کہ رویت کی حجت چھوڑوں اور ریڑھیاں بجھوا کر کھجور اور رسوئی چھوئے بغیر بیٹھوں

ریفرنڈم کے بگھار میں تغیر کیلئے نیل پھینٹیں اور فیسوں کی کالز ایکسچینج سولنگ کیلئے دیجئے

کولیشن رجسٹریشن کی استھان کیلئے روغنیات کی نقل پرنٹ کرکے لائیں

فاسٹ میں ڈرل کوچنگ کی لانگ فاؤل میں بیٹھیں اور اگلی منگل تک میاں سید سے صلح کریں

کموڈور خشونت کی چغلی افشا ہونے سے زوجہ کا توہم کافور ہوا

اخوان کے اجتماعات میں پہنچ کر رنگیلا مودی کی حیدر سے مڈھ بھیڑ ہوئی

کانجو گجر کی شیڈو ڈرائینگ میں کنویں سورج اور کوٹھری کو کھڑکی کے غلاف سے ججیں

حیف کہ دختر کی دکھے دل سے تجہیز کر سکے نہ بدھ کی ظہر تک سانسیں دبوچ سکیں

چوک کے نیون پر بیٹھی گیارہویں سفیر نے بیسویں قونصل کے شوالا کا کچوری کھوج لگایا

اوہو ہیلمٹ اور گھڑی شگوفوں کی آتشبازی دکھلا کر وین سے گریں

حرفوں کے ففٹی میچز سیکھے تو دشت سے نفوس کی پھوار اٹھی

تفاعل سے نفل پڑھیں مخفف اجر سے نہائیں اور دوغلے راگ سے زائد دہی ڈالیں

پچ کھیلنی ہے تو چھید میں ڈول پروئیں اور زیبرا بچھڑی کے سینگ پر سپننگ ڈبہ پھیریں

نیٹو ڈیبٹ کیس میں دوزخ کے سائے بزور مسخر کرنے کی فاش نقلیں ہیں

ساؤڑی ناخن چوسنے اور چگنے کے بعد لاؤڈ بیڑا میں دراڑیں ڈالنے آئی

واچ نگر کے خنازیر گوند نچوڑ کر دوروں پر آئے

میلے میں دودھیا چھاچھ تنہا خچر کیلئے کڑوا تھا

پھپھو کو کنوینس کرکے بیٹھنے کے بعد قوام کا نیڈل لگایئے

ہجویری تابعین کی سیرت اور سالمیت ان کے نفلی احیا کے رائج تک ایضا ہے

کوئل کو بھیرویں میچوں کی کھوکھلی گانٹھیں اڑنے سے پہلے سونپی تھیں

پٹرولیم کی جھیلیں تھم گئیں اور لیکوئڈ کینولا میگھا گوٹھ کے صدور تک پہنچیں

لوئر براڈ ویلز کا سیزن ڈھل کر ذیل کے ہجری دھائی تک کلوز ہوا

کھڈی اور ڈیش ہاؤسز کی اشیا چوہے اور انسیکٹی وورس کے معدہ میں سدھریں

نعم بخش کی اچھائی اوڈ ڈی لیفٹ بجھانے کا عشق ہے
بیلز سے سجی ایرو ویگن کا زیریں ہینڈ بجائیں اور سوئم و چہارم وید حفظ کریں
ہاں گل افگن زین کے کیفر کردار کی حد سے مفر نہیں
لنڈ ووڈ کا میوہ لوئی ویئرز کی باچھ سے گذاروں
مٹی کھود کر بچھائیں اور کوڑی کا تخم غور و خوض کے بعد بوئیں یا پھر سوئیں
اوئے برہان کے بقعہ اضلاع میں نہیں گھسنا
قذف لینے کیلئے بچھو پر حمل کا ڈاہا رگ دبائیں
کٹھن جھونکوں کی آندھیاں چلیں تو سیاروں کی تشبیہیں بگڑیں
پیروں نے جہاں چاہے مکعب انگوٹھیاں ارسال کیں تو گیندے کی تمحیص وقوع ہوئی
فلاں سیارے کے براعظم میں تعینات گوریلے کا نزول ہوا تو ایلچی کے رونگٹے کھڑے ہو گئے
فیض نے خصال راوی سے سگریٹ چھینی تو حج بخیر ہوا
دھوم سے ہوئے اول چہلم پر آقا کا تبصرہ خیر سے بھولیے
پگھلی بھیل کے داڑھ سے چمٹے بھوسے کی گٹھلی بھسم ہوئی
سانگی فلور وھیل کے پلڑے میں اگنے والے جھنڈ کا پچھلا آبیانہ ہیئت سے جعل نکلا
اولڈ خازن نوفل بورڈ نے یوزر کیلئے رزلٹ لوڈ نہیں کیا
روزے کا آخری عشرہ ہے ایک حصہ ڈونگا پکوڑوں کا تلوں
غشی سے پہلے سو سال کی بڑھیا نے کہا سونف اور اسبغول کے سفوف جھولی میں دو اور جئے جاؤ
انٹرمیڈیٹ شوز چلائیں تو فرقان ولد سرور اوسط درجے کا پگھلا ہے
کڑاہی کے چانپ گنواتے ہوئے جوزی نسل کے بیل پھڑپھڑائے
نشو نے پلو میں بڑ کا پرچہ گانٹھ کر ہموار ہینگ ٹھونسنے کی کوشش کی
آر جے بروری کے رویے پر سوچیں تو تعفن کا بھاؤ اڑائے
اپنی رہائش کی سیج کا نتیجہ کھوئے زوار نے انگشت جبڑے پر رکھ دی
نثار نے گھیا کی گٹھلی بوئی مگر بعد ازاں وہ صبوحی روگردانی کی ہیبت سے دہل گیا
صنف نسواں کو ٹیلی فیر ٹراؤزر والوں سے چھڑانے کیلئے ہائیڈروجن فلیگ سے مشقیں شیئر کریں
ٹھوڑی اور ایڑی پر چیر لگایا تو سحر تیزی سے اچھلی
چولی کا جوڑا پہنتے ہوئے بیٹھو تو لائیو اپھارہ برسنے میں ٹھہراؤ آ جاتا ہے
گدا چشم جیش کو گھر کی ڈھارس ہوئی تو وہ لون پر بچھی دھار پر سویا
بیف اور چھوہارے کی لوٹ پر ہیجڑے کا ناچ نچا کر بوائز کے جوہر معلوم کئے
ریڑھ سے ماؤتھ کی جڑوں تک این ایم نینی ردیف کے ایز سے نکھر گئی
نیور بی فول خواہ ایگزٹ کے کیو میں ملوں یا رئیل پلگ تک آؤں
جو فلک بوس ای ون چئر بخشے اسے کہاں سے لاؤں گدھو

سوئفٹ گیند کے سرخیل وہپ فشر سے بنیادی میسج کا شوشہ پوچھو
سوک کوئز کے بد سلوک سے آنکھیں کھولیں تو حب اغراض سے چیخ مار کر توجہ مبذول کرائی
ہلکی پھلکی چادر بچھ بچھا کر سنچریاں گذاروں اور دیگچی کے بیخ میں نہاں مونگ کے پراٹھے داغ جاؤں
بوجھو تو بھڑ جیسا کیڑا کیا جانوں جو غول بنائے اور سوسو فصلیں کچل کر چھائے
یہ گتھیاں نہ پڑھائیں کہ شعلے کو چھوا نہیں جاتا
کسی کی دلجوئی کیلئے اپنا جگر دھویا اور ہندسے ٹھونسا کر معینات میں ٹھونس دیئے
بد خو نثار نے کہا کہ میں اسے بھوسہ بھرے ہتھ گاڑی کے بہی کھاتے دلواؤں
میں نے حق کے جویا سے کہا کہ وقائع پر بھنا کر رائے مت دو
ترن کا چوڑا شش پایہ کھر صاحب کو جچے اور اسے ہی چاق و چوبند بنائے
کھوئی کی جانب کچھے جانے کو کیا روؤں اگر کھنچ جاؤں تو پانی میں گھل جاؤں
کیلشم فاسفیٹ کے آمیزے کو مصنوعی مسوڑوں کے بنانے کیلئے فراہم کیا جائے گا
بلور کاؤچ سے سویلین تمغے ڈیزائن کرنے کا کام ہو گا
بارہواں ہمزاد دوربین سے شفاف بائیوس کا جائزہ لینے لگا
نجف کے فیوض و افکار کی چھڑی میں بدائع کے عیب یعنی بےحرمتی کا مت سوچو
گوشت اور پھوگ کی تقلیل میں تغافل کی سوجی چھڑک کر وثوق سے اُٹھو
مضمون بعنوان بونوں کو غزل کی تہمت ناچیز سے کہلوا کر معزز قوتوں کو سمجھایا
غیرفیصلہ عقوبت کی رود میں ہر ایک کو شاذ و نادر بخشو
پیراتھرائیڈ سے تولیدی جین کے پھیپھڑوں کا غبار ظاہر ہوتا ہے
چوتھا ایوننگ سیشن متعینہ زاویوں کے انضباطی پھیلاؤ کا پراسیس ہے
بوجہ غبن مائیکروویوز ایجاد پر لاگت آٹھ ہندسوں سے بڑھ گئی
بجز خفیف غیظ کے آنکھیں دکھیں تو مت میچو
ماضی کو بھلایا ہے تو آگے بڑھیں یونہی گناہوں میں نہ پڑیں
جنگ چھڑنے سے پہلے بروے نے مجھے کچھ آڑھا سیدھا طریقہ اور مشق سمجھائے

Appendix D

Phone-frequency comparison of Corpus and Sentences

	Phones (CISAMPA)	Phones (IPA)	Frequency in Sentences	Frequency in Corpus
1	**A**	ə	3987	21901145
2	**AA**	ɑ	3447	17908279
3	**R**	r	2725	12147033
4	**K**	k	2128	11126497
5	**AE**	e	2586	10211039
6	**I**	ɪ	1543	9774520
7	**N**	n	1848	9459014
8	**M**	m	1626	9214641
9	**II**	i	2108	8690981
10	**S**	s	1545	7871940
11	**H**	h	1206	7504548
12	**T_D**	t	992	6765553
13	**L**	t̪	1433	6640219
14	**U**	ʊ	959	5844892
15	**D_D**	d̪	902	4174674
16	**B**	b	973	3838562
17	**OO**	o	1016	3423191
18	**UU**	u	667	3378480
19	**J**	j	638	3251417
20	**P**	p	510	2708848
21	**D_ZZ**	ʤ	601	2584718
22	**Z**	z	681	2459025
23	**V**	v	623	2303941
24	**AY**	æ	432	1858386
25	**F**	f	531	1760538
26	**O**	ɔ	570	1718142
27	**Q**	q	345	1658061
28	**G**	ɡ	582	1652945
29	**AEN**	ẽ	589	1624624

30	**SH**	ʃ	563	1611315
31	**AYN**	æ̃	70	1609563
32	**TT**	t	274	1488132
33	**X**	χ	329	1063899
34	**T_SH**	ʧ	407	1035146
35	**OON**	õ	322	958383
36	**E**	ɛ	152	660513
37	**DD**	d	299	550995
38	**K_H**	k^h	217	488610
39	**RR**	ɽ	249	468696
40	**B_H**	b^h	123	462408
41	**T_D_H**	t̪^h	157	456386
42	**IIN**	ĩ	87	412644
43	**7**	ɣ	201	278670
44	**NG**	ŋ	161	276152
45	**T_SH_H**	ʧ^h	135	249198
46	**AAN**	ɑ̃	125	195061
47	**UUN**	ũ	59	164538
48	**D_D_H**	d̪^h	105	117399
49	**TT_H**	t^h	126	115661
50	**D_ZZ_H**	ʤ^h	92	114620
51	**P_H**	p^h	54	108979
52	**RR_H**	ɽ^h	57	96715
53	**G_H**	g^h	62	70057
54	**DD_H**	d^h	33	11912
55	**ZZ**	ʒ	18	10164
56	**Y**	ʔ	9	9573
57	**N_H**	n^h	7	4017
58	**ON**	ɔ̃	1	240
59	**R_H**	r^h	1	111
60	**V_H**	v^h	1	64

Appendix E

Interview Questions (Sample)

Part-I

- What is your name?
- What is your gender?
- What is your height?
- What is your place of birth?
- What is your date of birth? Please answer using the format پانچ نوبمر انیس سو چالیس
- Which is your current area of residence?
- Which, if any, are some other areas of your previous residences?
- Which school or schools did you attend?
- Which other educational institutes have you attended, if any?
- What is your current profession?

Part-II

- Explain the route you took to get from your home to this location.
- How long does it take to get to work every day?
- What are your responsibilities at work?
- Describe a normal day at work.
- How did your day go yesterday? Describe with timelines if possible.
- Describe a memorable day.
- Describe a funny experience.
- Describe a scary experience.
- Describe an interesting experience.
- What is your greatest fear?
- Which places would you like to visit and why?
- Describe an accomplishment that you are proud of.

- Do you have any brothers or sisters? Are they younger or older than you?
- Tell us about your friends.
- Name some of your closest friends.
- How did you become friends?
- Tell us about three of your favorite childhood memories.

Extension to Part-II (In case the answers are short)

- What do you do in your free time?
- Where do you normally go for dining out and why?
- What is your favorite food?
- What is your least favorite food?
- Do you enjoy watching films?
  - What types of films do you watch?
  - Which are some of your favorite films?
  - Describe your favorite character from one of these films.
  - Which is the last film you watched?
  - How was the last film you watched?
- Do you enjoy reading books?
  - What types of books do you read?
  - Which are some of your favorite books?
  - Name some of your favorite writers.
  - Which is the last book you read?
  - How was the last book you read?
- Do you enjoy listening to music?
  - What type of music do you listen to?
  - Who are some of your favorite musicians?
  - What are you listening to at currently?
- What is your favorite television channel?
- Which programs do watch on television?
- Which is your favorite program on television?

- Are you interested in cricket or any other sport?
- Describe a cricket match which you cannot forget.
- Which newspaper(s) do you read regularly?
- Describe your favorite type of weather.
- Where do you go to shop?
- What are your future goals?
- Describe any interesting news that you saw on TV lately or read about.
- Name an event which you consider was a turning point in Pakistan's history? And explain why you think so.
- What was your reaction when Pakistan won the world cup? What were you doing then?
- What were you doing when 9/11 took place? What was your reaction?

Appendix F

Report and Frequency Files Generated by Sphinx Files Compiler

Sample Report File

```
Test Report Generated on: Sat Jun 27 04:01:19 VET 2009

No of Sentences/Utterances in the training file:  1685
No of Sentences/Utterances in the testing file:   50

No of words in the training file:   10677
No of words in the testing file:  313 i.e. 2% of Training Data

No of Unique words in the training file:   1284
No of Unique words in the testing file:  125 i.e. 9% of Training Data

No of Phones in the training file:  36998
No of Phones in the testing file:   1172

No of Unique Phones in the training file:  57
No of Unique Phones in the testing file:     46

No of Unique overlapping words between the Training and test data: 113
No of Unique overlapping Phones between the Training and test data:   46

No of Unique non-overlapping words between the Training and test data:  12
No  of  Unique  non-overlapping  Phones  between  the  Training  and  test  data:
  0

No of Overlapping Words occurring in the test data:  301
No of Overlapping Phones occurring in the test data:   1172

No of Non-Overlapping Words occurring in the test data:   12
No of Non-Overlapping Phones occurring in the test data:  0

Phone to Frequency for training File written to:   Sphinx\Test1\TrP2Fr.txt
Phone to Frequency for testing  File written to:   Sphinx\Test1\TeP2Fr.txt
Word to Frequency for training File written to:    Sphinx\Test1\TrW2Fr.txt
Word to Frequency for testing File written to:  Sphinx\Test1\TeW2Fr.txt
```

Sample Word-Frequency listing for Training File (Partial)

```
LOOGOON     7
KOONSAA     2
OSAT_DAN      3
SAVAALAAT_D      3
KAHAA         9
VAAQIII     10
DDIIBAETTS       1
HUUII         24
VUZARAA     1
SALAAIIDDZ       1
MA7RIB      2
MIZAAHIJAA       6
BAERUUNII     1
XUSUSAN     1
D_ZZAAT_DII      3
AD_DAA      2
ZEHMAT_D      1
TTIDDDDIJAAN     1
MAAIKROOPAROOSAYSAR1
T_SH_HUUTTII     3
KOOSHISH      12
ZAEHN         5
T_SHILLAA     1
MAAR        1
SAST_DAANAE      1
AEHSAAS     2
ULD_ZZ_HAN       2
MAAN        1
SARRKAEN      3
D_DAAXILAE       1
GII         11
PARR_HAAAE       1
PARR_HAAT_DAA    3
PAT_DT_DAA       4
D_DAAXILAA       8
ZARAA         1
PAYNT_DAALIIS    1
AMUUMII     2
BAIID_D     1
KAALID_ZZ     10
UT_DAAR     2
T_DAD_DRIIS      5
XUSUUSII      1
B_HAII      1
D_DAER      3
XART_SHAA     2
D_DAEN      1
D_DOPEHR      1
```

Sample Phone-Frequency listing for Training File (Complete)

```
G_H       52
Z      469
Y      1
AE     2946
X      133
DD     77
V      649
AA     3174
U      522
S      1700
AEN       575
R      2163
Q      242
SH     207
P      713
O      524
N      1538
M      1553
L      870
K      2271
J      626
I      1239
H      1731
OON       112
G      318
F      285
E      166
DD_H      1
B      666
A      3414
T_SH_H    137
UU     491
RR     111
T_D_H       295
OO     917
RR_H      43
D_D       635
II     1691
7      32
AAN       96
K_H       46
T_D       1265
B_H       128
UUN       114
D_ZZ      509
AYN       172
T_SH      310
ZZ     1
TT     416
```

```
R_H       3
D_D_H       11
NG     39
D_ZZ_H    108
AY     331
IIN       104
P_H       31
TT_H      25
```

Sample Word-Frequency listing for Test File (Partial)

```
POOLOO      1
HUUN        1
INT_SH      1
T_DAARIIX      1
PEHLAE      1
AAF         1
PIT_DT_DAA       1
AKAYDDMII      1
PURAANAA       1
NAAM        1
PAAS        1
MAERAA      3
RAHAE          1
JANII          3
AE        1
SO        2
BAYT_SHALAR      1
PARR_HII       1
RIIPIITT       1
KENTT          2
SII         1
AND_DAR     1
HISSAA      1
D_DASVAEN      1
MAYRII      5
T_DOR          1
KAALUUNII      1
D_ZZAMAAAT_D     5
T_DAARIIXAE      1
T_D_HAA     3
GARAAUNDD      1
SAAT_DVAEN       1
T_SH_HATTII      2
T_DAVIIL       1
GAYRAEZAN      2
AVVAL          2
NASHAAT_D      1
ES        1
MAEN        21
AEK         14
```

Sample Phone-Frequency listing for Test File (Complete)

```
G_H       5
Z       7
AE      92
X       2
DD      9
V       22
AA      124
U       7
S       54
AEN       24
R       82
Q       16
SH      13
P       28
O       22
N       50
M       56
L       49
K       75
J       11
I       50
H       49
OON       2
G       6
F       8
E       14
B       13
A       88
T_SH_H    2
UU      10
T_D_H       4
OO      9
RR_H      1
D_D       19
II      55
7       1
T_D       33
UUN       1
D_ZZ      20
T_SH      3
TT      15
R_H       2
NG      1
AY      16
IIN       1
TT_H      1
```

Appendix G

Setting up CMU Sphinx Speech Recognition System in Windows®

Disclaimer

This section describes in detail the procedure to set up CMU Sphinx trainer and decoder on an Microsoft Windows® based system. These details have been largely extracted and compiled from the tutorials, manuals and discussion blogs mentioned in the references. I have filled in the gaps and added details where required. Some of the discussed details deal with Urdu Specific problems.

System Specifications

This procedure has been tested on a *Toshiba* Notebook computer (*Toshiba Satellite M115*), with a 1.6 GHz dual core processor (*T2050*), 2.5 GB of RAM and 80 GB conventional Hard Disk Drive. The Operating System is *Genuine Microsoft Windows XP, Media Center Edition, Version 2002*, and *Service Pack 3*. Primary IDEs used in this procedure are *Microsoft Visual Studio 6.0*, *Microsoft Visual Studio 2008* and *Eclipse SDK version 3.4.2* (Ganymede). The system is running JRE 1.6.

Introduction

This walkthrough has been designed to facilitate setting up a Sphinx based recognition system. The material used in this tutorial has been extracted/taken/derived from different tutorials and helping materials available at ([8], [13], [9], [10], [11], [54], [55], [56] and [57]). In addition it has been tested and modified by the training and testing of speaker specific medium vocabulary continuous and spontaneous automatic speech recognition system for Urdu. Urdu digit recognition system development is taken as the design goal for this system.

Software setup[5]

1. Perl

Install ActivePerl for Windows, which is available from ActiveState[6].

[5] Extracted from: Robust Group Tutorial, http://www.speech.cs.cmu.edu/sphinx/tutorial.html#app1

2. C Compiler

Install Microsoft Visual C++ 6.0 for Sphinx Train and MS VC++ 2008 for Sphinx-III and Sphinx Base (I have used Sphinx3 and Sphinx4 both for decoding purposes).

3. Java Platform

For Sphinx-4 install JDK and Eclipse.

Setting up the data

Download AN4[7] speech database (this will be modified for our own Speech recognition system). AN4 includes the audio, but it is a very small database and we are not interested in the audio as we will provide our own.

The steps involved:

Create a directory for the system e.g. *tutorial*, and move to that directory.

1. Download the audio tarball AN4 and save it to the same tutorial directory you just created.
2. In Windows, using the Windows Explorer, go to the tutorial directory, right-click the tarball, and choose "Extract to here" in the WinZip menu.

By the time you finish this, you will have a tutorial directory with the following contents

Tutorial

- an4
- an4_sphere.tar.gz

Setting up the trainer

Code retrieval

SphinxTrain can be retrieved by downloading its compressed version.

[6] ActiveState: http://www.activestate.com/Products/activeperl/index.mhtml

[7] AN4: http://www.speech.cs.cmu.edu/databases/an4/ Download from: http://www.speech.cs.cmu.edu/databases/an4/an4_sphere.tar.gz

- Using the tarball, download the SphinxTrain tarball[8] by clicking on the link and choosing "Save" when the dialog window appears. Save it to the same tutorial directory. Extract the contents as follows.
    - In Windows, using the Windows Explorer, go to the tutorial directory, right-click the SphinxTrain tarball, and choose "Extract to here" in the WinZip menu.

By the time you finish this, you will have a tutorial directory with the following contents

Tutorial

- an4
- an4_sphere.tar.gz
- SphinxTrain
- SphinxTrain.nightly.tar.gz

Compilation

In Windows:

1. Double click the file tutorial/SphinxTrain/SphinxTrain.sln. This will open MS Visual C++ (use version 6.0).
2. In the Menu Build choose Batch Build, and select all items. Click on Rebuild All This will build all executables needed by the trainer.

Tutorial Setup

After compiling the code, you will have to setup the tutorial by copying all relevant executables and scripts to the same area as the data. Assuming your current working directory is tutorial, you will need to do the following.

```
cd SphinxTrain
```

# If you installed AN4

perl scripts_pl/setup_tutorial.pl an4

[8] Sphinx Train: http://cmusphinx.org/download/nightly/SphinxTrain.nightly.tar.gz

Setting up the Training data

I followed the following steps for a simple Urdu digit recognition system:

1. Place all training audio files in the (*an4\wav\{name your training folder}*) folder. Use either wav (mswav), .sph or .nist format. Praat supports all these formats. I used nist. Record some separate utterances e.g. AIK, DO,… and some combined utterances e.g. counting from SIFAR till DAS. Collect enough training data e.g. I collected 5 minutes of recordings in my voice for the 11 digits from SIFAR to DAS.
2. Now in the an4\etc\ folder, in the an4.dic file, define the utterance to phone mappings. You may map at word or phone level. e.g. you may map utterance *AIK* to a single phone *AIK,* or to phone *AY K,* or define any other useful substitute. For small vocabularies, word level mapping are preferred while phone level mappings are preferred for larger vocabularies. More than one pronunciation mappings can be shown with a (1) and (2) etc. after the word.

   e.g.

AIK	AE K
AIK(1)	AY K

3. In the filler dictionary in the etc folder (an4.filler), define the non-speech utterances i.e. the start of utterance silence <s>, the end of utterance silence <\s> and the middle of utterance silence <sil>. Map them all to the same phone SIL, which models silence or the background noise.
4. In the phone file define all the phones including the silence SIL as follows. There should be no empty lines
5. In the file an4*_train.fileids,* define all audio file ids without extensions with references to the root (wav\an4_train\) folder.
6. In the *an4_train.transcription* file establish utterance to audio mappings. Remember, these are not phone to audio mappings but mappings between the words in the left column of the dictionary file and the audio files. Important to note is that the files should be in the same order as described in the *an4_train.fileids* file.

7. In the file an4_*test.fileids,* define all test data audio file ids without extensions with references to the root (wav\an4_test\) folder.

8. In the *an4_test.transcription* file establish utterance to audio mappings. Remember, these are not phone to audio mappings but mappings between the words in the left column of the dictionary file and the audio files. Important to note is that the files should be in the same order as described in the *an4_test.fileids* file.

9. Now make the necessary changes in the *sphinx_train.cfg* file. My file is shown below with the modified and/or potentially modified areas shown in bold:

```
# Configuration script for sphinx trainer                  -*-mode:Perl-*-

$CFG_VERBOSE = 1;     # Determines how much goes to the screen.

# These are filled in at configuration time
$CFG_DB_NAME = "an4";
$CFG_BASE_DIR = "E:/Sphinx/an4";
$CFG_SPHINXTRAIN_DIR = "E:/Sphinx/SphinxTrain";

# Directory containing SphinxTrain binaries
$CFG_BIN_DIR = "$CFG_BASE_DIR/bin";
$CFG_GIF_DIR = "$CFG_BASE_DIR/gifs";
$CFG_SCRIPT_DIR = "$CFG_BASE_DIR/scripts_pl";

# Experiment name, will be used to name model files and log files
$CFG_EXPTNAME = "$CFG_DB_NAME";

# Audio waveform and feature file information
$CFG_WAVFILES_DIR = "$CFG_BASE_DIR/wav";
$CFG_WAVFILE_EXTENSION = 'nist';
$CFG_WAVFILE_TYPE = 'nist'; # one of nist, mswav, raw
$CFG_FEATFILES_DIR = "$CFG_BASE_DIR/feat";
$CFG_FEATFILE_EXTENSION = 'mfc';
$CFG_VECTOR_LENGTH = 13;

$CFG_MIN_ITERATIONS = 1;  # BW Iterate at least this many times
$CFG_MAX_ITERATIONS = 10; # BW Don't iterate more than this, somethings
likely wrong.

# (none/max) Type of AGC to apply to input files
$CFG_AGC = 'none';
# (current/none) Type of cepstral mean subtraction/normalization
# to apply to input files
$CFG_CMN = 'current';
# (yes/no) Normalize variance of input files to 1.0
$CFG_VARNORM = 'no';
```

```
# (yes/no) Use letter-to-sound rules to guess pronunciations of
# unknown words (English, 40-phone specific)
$CFG_LTSOOV = 'no';
# (yes/no) Train full covariance matrices
$CFG_FULLVAR = 'no';
# (yes/no) Use diagonals only of full covariance matrices for
# Forward-Backward evaluation (recommended if CFG_FULLVAR is yes)
$CFG_DIAGFULL = 'no';

# (yes/no) Perform vocal tract length normalization in training.  This
# will result in a "normalized" model which requires VTLN to be done
# during decoding as well.
$CFG_VTLN = 'no';
# Starting warp factor for VTLN
$CFG_VTLN_START = 0.80;
# Ending warp factor for VTLN
$CFG_VTLN_END = 1.40;
# Step size of warping factors
$CFG_VTLN_STEP = 0.05;

# Directory to write queue manager logs to
$CFG_QMGR_DIR = "$CFG_BASE_DIR/qmanager";
# Directory to write training logs to
$CFG_LOG_DIR = "$CFG_BASE_DIR/logdir";
# Directory for re-estimation counts
$CFG_BWACCUM_DIR = "$CFG_BASE_DIR/bwaccumdir";
# Directory to write model parameter files to
$CFG_MODEL_DIR = "$CFG_BASE_DIR/model_parameters";

# Directory containing transcripts and control files for
# speaker-adaptive training
$CFG_LIST_DIR = "$CFG_BASE_DIR/etc";

#*******variables used in main training of models*******
$CFG_DICTIONARY     = "$CFG_LIST_DIR/$CFG_DB_NAME.dic";
$CFG_RAWPHONEFILE   = "$CFG_LIST_DIR/$CFG_DB_NAME.phone";
$CFG_FILLERDICT     = "$CFG_LIST_DIR/$CFG_DB_NAME.filler";
$CFG_LISTOFFILES    = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.fileids";
$CFG_TRANSCRIPTFILE = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.transcription";
$CFG_FEATPARAMS     = "$CFG_LIST_DIR/feat.params";

#*******variables used in characterizing models*******

$CFG_HMM_TYPE = '.cont.'; # Sphinx III
#$CFG_HMM_TYPE  = '.semi.'; # Sphinx II

if (($CFG_HMM_TYPE ne ".semi.") and ($CFG_HMM_TYPE ne ".cont.")) {
  die "Please choose one CFG_HMM_TYPE out of '.cont.' or '.semi.', " .
    "currently $CFG_HMM_TYPE\n";
}
```

```
if ($CFG_HMM_TYPE eq '.semi.') {
  $CFG_DIRLABEL = 'semi';
  $CFG_STATESPERHMM = 5;
  $CFG_SKIPSTATE = 'yes';
# Four (4) stream features for Sphinx II
  $CFG_FEATURE = "s2_4x";
  $CFG_NUM_STREAMS = 4;
  $CFG_INITIAL_NUM_DENSITIES = 256;
  $CFG_FINAL_NUM_DENSITIES = 256;
  die "For semi continuous models, the initial and final models have the
same density"
    if ($CFG_INITIAL_NUM_DENSITIES != $CFG_FINAL_NUM_DENSITIES);
} elsif ($CFG_HMM_TYPE eq '.cont.') {
  $CFG_DIRLABEL = 'cont';
  $CFG_STATESPERHMM = 3;
  $CFG_SKIPSTATE = 'no';
# Single stream features - Sphinx 3
  $CFG_FEATURE = "1s_c_d_dd";
  $CFG_NUM_STREAMS = 1;
  $CFG_INITIAL_NUM_DENSITIES = 1;
  $CFG_FINAL_NUM_DENSITIES = 8;
  die "The initial has to be less than the final number of densities"
    if ($CFG_INITIAL_NUM_DENSITIES > $CFG_FINAL_NUM_DENSITIES);
}

# (yes/no) Train multiple-gaussian context-independent models (useful
# for alignment, use 'no' otherwise) in the models created
# specifically for forced alignment
$CFG_FALIGN_CI_MGAU = 'no';
# (yes/no) Train multiple-gaussian context-independent models (useful
# for alignment, use 'no' otherwise)
$CFG_CI_MGAU = 'no';
# Number of tied states (senones) to create in decision-tree clustering
$CFG_N_TIED_STATES = 1000;
# How many parts to run Forward-Backward estimatinon in
$CFG_NPART = 1;

# (yes/no) Train a single decision tree for all phones (actually one
# per state) (useful for grapheme-based models, use 'no' otherwise)
$CFG_CROSS_PHONE_TREES = 'no';

# Use force-aligned transcripts (if available) as input to training
$CFG_FORCEDALIGN = 'no';

# Use a specific set of models for force alignment.  If not defined,
# context-independent models for the current experiment will be used.
$CFG_FORCE_ALIGN_MDEF                                                   =
"$CFG_BASE_DIR/model_architecture/$CFG_EXPTNAME.falign_ci.mdef";
if ($CFG_FALIGN_CI_MGAU eq  'yes') {
  $CFG_FORCE_ALIGN_MODELDIR                                             =
"$CFG_MODEL_DIR/$CFG_EXPTNAME.falign_ci_${CFG_DIRLABEL}_$CFG_FINAL_NUM_DEN
SITIES";
}
else {
  $CFG_FORCE_ALIGN_MODELDIR                                              =
"$CFG_MODEL_DIR/$CFG_EXPTNAME.falign_ci_$CFG_DIRLABEL";
}

# Use a specific dictionary and filler dictionary for force alignment.
# If these are not defined, a dictionary and filler dictionary will be
# created from $CFG_DICTIONARY and $CFG_FILLERDICT, with noise words
# removed from the filler dictionary and added to the dictionary (this
# is because the force alignment is not very good at inserting them)

#                       $CFG_FORCE_ALIGN_DICTIONARY                      =
"$ST::CFG_BASE_DIR/falignout$ST::CFG_EXPTNAME.falign.dict";;
#                       $CFG_FORCE_ALIGN_FILLERDICT                      =
"$ST::CFG_BASE_DIR/falignout/$ST::CFG_EXPTNAME.falign.fdict";;

# Use a particular beam width for force alignment.  The wider
# (i.e. smaller numerically) the beam, the fewer sentences will be
# rejected for bad alignment.
$CFG_FORCE_ALIGN_BEAM = 1e-60;

# Calculate an LDA/MLLT transform?
$CFG_LDA_MLLT = 'no';
# Dimensionality of LDA/MLLT output
$CFG_LDA_DIMENSION = 29;

#set convergence_ratio = 0.004
$CFG_CONVERGENCE_RATIO = 0.04;

# Queue::POSIX for multiple CPUs on a local machine
# Queue::PBS to use a PBS/TORQUE queue
$CFG_QUEUE_TYPE = "Queue";

# Name of queue to use for PBS/TORQUE
$CFG_QUEUE_NAME = "workq";

# (yes/no) Build questions for decision tree clustering automatically
$CFG_MAKE_QUESTS = "yes";
# If CFG_MAKE_QUESTS is yes, questions are written to this file.
# If CFG_MAKE_QUESTS is no, questions are read from this file.
$CFG_QUESTION_SET                                                        =
"${CFG_BASE_DIR}/model_architecture/${CFG_EXPTNAME}.tree_questions";
#$CFG_QUESTION_SET = "${CFG_BASE_DIR}/linguistic_questions";

$CFG_CP_OPERATION                                                        =
"${CFG_BASE_DIR}/model_architecture/${CFG_EXPTNAME}.cpmeanvar";

# This variable has to be defined, otherwise utils.pl will not load.
```

```
$CFG_DONE = 1;

return 1;
```

Training the ASR System

Go to the directory where you installed the data.

```
cd ..\an4
```

The system does not directly work with acoustic signals. The signals are first transformed into a sequence of feature vectors, which are used in place of the actual acoustic signals. To perform this transformation (or parameterization) from within the directory an4, type the following command on the command line.

```
perl scripts_pl\make_feats.pl  -ctl etc\an4_train.fileids
```

This script will compute, for each training utterance, a sequence of 13-dimensional vectors (feature vectors) consisting of the Mel-frequency cepstral coefficients (MFCCs). Note that the list of wave files contains a list with the full paths to the audio files. Since the data are all located in the same directory as you are working, the paths are relative, not absolute. You may have to change this, as well as the *an4_test.fileids* file, if the location of data is different. The MFCCs will be placed automatically in a directory called .\feat\ in an4.

Now, simply run the RunAll.pl script provided.

```
perl scripts_pl\RunAll.pl
```

One of the files that appear in your current directory is an .html file, named an4.html. This file will contain a status report of jobs already executed. Verify that the job you launched completed successfully.

You have now completed your training. The final models and their locations will depend on the database and the model type that you are using. If you are using AN4 to train continuous models, you will find the parameters of the final 8 Gaussian\state 3-state CD-tied acoustic models (HMMs) with 1000 tied states in a directory called .\model_parameters\an4.cd_cont_1000_8\. You will also find a model-index file for these models called an4.1000.mdef in .\model_architecture\. This file is used by the system to associate the appropriate set of HMM parameters with the HMM for each sound unit you are modeling.

Running the Sphinx-3 decoder (Batch testing the ASR System)

There are a few problems running the Sphinx-3 decoder out-of-the-box. Ideally the process should follow after the training step as follows:

Setting up the decoder

SPHINX-3

- **Code retrieval**

SPHINX-3 can be downloaded as a tarball. It is also available as a release from SourceForge.net. Using the tarball, download the sphinx3 tarball and sphinxbase by clicking on the link and choosing "Save" when the dialog window appears. Save them to the same tutorial directory. Extract the contents as follows.

- tutorial
    - an4
    - SphinxTrain
    - sphinx3
    - sphinxbase

- **Compilation**

In Windows, if you download SphinxBase from the release system, please rename it (e.g. from 'sphinxbase-0.1') to 'sphinxbase' and then:

1. Double click the file tutorial/sphinxbase/sphinxbase.sln. This will open MS Visual C++, if you have it installed.
2. In the Menu Build choose Batch Build, and select all items. Click on Rebuild All This will build all libraries in the SphinxBase package.

3. Double click the file tutorial/sphinx3/programs.sln. This will open MS Visual C++, if you have it installed.
4. In the Menu Build choose Batch Build, and select all items. Click on Rebuild All This will build all executables in the SPHINX-3 package.

- **Tutorial Setup**

After compiling the code, you will have to setup the tutorial by copying all relevant executables and scripts to the same area as the data. Assuming your current working directory is tutorial, you will need to do the following.

```
cd sphinx3
```

```
perl scripts/setup_tutorial.pl an4
```

Running the Sphinx-3 Decoder

Decoding is relatively simple to perform. First, compute MFCC features for all of the test utterances in the test set. To compute MFCCs from the wave files, from the top level directory, namely an4, type the following from the command line:

```
perl scripts_pl/make_feats.pl  -ctl etc/an4_test.fileids
```

You are now ready to decode. Type the command below.

```
perl scripts_pl/decode/slave.pl
```

When you run the decode script, it will print information about the accuracy in the top level .html page for your experiment. It will also create two sets of files. One of these sets, with extension .match, contains the hypothesis as output by the decoder. The other set, with extension .align, contains the alignment generated by your alignment program, or by the built-in script, with the result of the comparison between the decoder hypothesis and the provided transcriptions. If you used the NIST tool, the .html file will contain a line such as the following if you used an4:

```
SENTENCE ERROR: 56.154% (73/130)   WORD ERROR RATE: 16.429% (127/773)
```

The second percentage number is the WER and has been obtained using the 8 Gaussians per state HMMs that you have just trained in the preliminary training run. Other numbers in the above output will be explained later in this document. The WER may vary depending on which decoder you used.

IMPORTANT NOTES: However, the problem occurs when you execute "**perl scripts_pl/decode/slave.pl**" as the perl script to setup the sphinx3 tutorial fails to make the "sphinx_decode.cfg" in the an4\etc\ folder. So follow the procedure given below:

- In cmd, do *cd an4*
- Run > perl ../sphinx3/scripts/setup_sphinx3.pl -task an4 (this will create the "sphinx_decode.cfg" in the an4\etc\)
- Now setup the Sphinx-3 tutorial again (this is just in case, I haven't tried otherwise yet).

```
cd sphinx3
perl scripts/setup_tutorial.pl an4
```

- Reconfigure the sphinx_decode.cfg file as before, and configure the sphinx_decode.cfg file if required.
- Now it will start giving you a lot of error when you will run the **perl scripts_pl/decode/slave.pl,** so copy the following to the an4\bin\:
  - sphinx3_decode.exe (from: \sphinx3\bin\Release)
  - sphinxbase.dll (from: sphinxbase\lib\Release)
  - s3decoder.dll (from: sphinx3\lib\Release)
- Moreover you need a language model in the form of a binary dump file in the an4\etc\ folder
- So, download: **lm3g2dmp**, from:

  http://cmusphinx.org/download/nightly/lm3g2dmp.nightly.tar.gz
- Unzip and build with VC 6. Use Project>Rebuild All
- Now in the Release directory find **lm3g2dmp.exe**
- This converts lm to dmp. On cmd prompt > lm3g2dmp lmFileName.lm DestDirectory

    - e.g. lm3g2dmp an4.lm .\

- Now you need an lm file. So give the text corpus to the online LM tool:

http://www.speech.cs.cmu.edu/tools/lmtool-adv.html

If the number of tokens in the corpus is greater than 5000, the online LM toolkit will not work and the Statistical Language Modeling Toolkit needs to be used. Please see the next section.

- Save the resulting lm file and convert to dmp. Then give the path in the lm entry in the sphinx_decode.cfg file.
- Now if all went well, **perl scripts_pl/decode/slave.pl,** should work. See the an4-1-1.match, an4.align, an4.match files in the an4\result\.

Using the Statistical Language Modeling Toolkit to create Language Models

The SLM toolkit works only on Unix based systems. Download the toolkit and before building it, for "little-endian" machines the variable BYTESWAP_FLAG will need to be set in the Makefile. This can be done by editing src/Makefile directly, so that the line

```
#BYTESWAP_FLAG  = -DSLM_SWAP_BYTES
```

is changed to

```
BYTESWAP_FLAG  = -DSLM_SWAP_BYTES
```

Then simply change to the src/ directory and perform:

```
make install
```

The executables will then be copied into the bin/ directory, and the library file SLM2.a will be copied into the lib/ directory.

To convert an ASCII corpus file into a trigram language model, with Witten-Bell discounting, place it into the /bin directory and perform the following steps:

- ./text2wfreq <Corpus.txt> a.wfreq
- ./wfreq2vocab <a.wfreq> a.vocab

- ./text2idngram -n 3 -vocab a.vocab <Corpus.txt> a.idngram
- ./idngram2lm -n 3 -vocab_type 2 -witten_bell -oov_fraction 0.5 -idngram a.idngram -vocab a.vocab -arpa LanguageModel.arpa

-linear | -absolute | -good_turing | -witten_bell switches can be used for other smoothing schemes. Please see the toolkit documentation for more details.

Modifying the Beam Width, Language Weight etc

In order to modify these parameters, the sphinx_decode.cfg file in the etc folder needs to be edited. Following is the complete decode configuration file with the parameters that I modified shown in bold:

```
# Configuration script for sphinx decoder                  -*-mode:Perl-*-

# Variables starting with $DEC_CFG_ refer to decoder specific
# arguments, those starting with $CFG_ refer to trainer arguments,
# some of them also used by the decoder.

$DEC_CFG_VERBOSE = 1;   # Determines how much goes to the screen.

# These are filled in at configuration time
$DEC_CFG_DB_NAME = 'an4';
$DEC_CFG_BASE_DIR = 'E:/Sphinx/an4';
$DEC_CFG_SPHINXDECODER_DIR = 'E:/Sphinx/sphinx3';
$DEC_CFG_SPHINXTRAIN_CFG = "$DEC_CFG_BASE_DIR/etc/sphinx_train.cfg";

# Name of the decoding script to use (psdecode.pl or s3decode.pl,
probably)
$DEC_CFG_SCRIPT = 's3decode.pl';

require $DEC_CFG_SPHINXTRAIN_CFG;

$DEC_CFG_BIN_DIR = "$DEC_CFG_BASE_DIR/bin";
$DEC_CFG_GIF_DIR = "$DEC_CFG_BASE_DIR/gifs";
$DEC_CFG_SCRIPT_DIR = "$DEC_CFG_BASE_DIR/scripts_pl";

$DEC_CFG_EXPTNAME = "$CFG_EXPTNAME";
$DEC_CFG_JOBNAME  = "$CFG_EXPTNAME"."_job";

# Models to use.
$DEC_CFG_MODEL_NAME =
"$CFG_EXPTNAME.cd_${CFG_DIRLABEL}_${CFG_N_TIED_STATES}";

$DEC_CFG_FEATFILES_DIR = "$DEC_CFG_BASE_DIR/feat";
$DEC_CFG_FEATFILE_EXTENSION = '.mfc';
```

```
$DEC_CFG_VECTOR_LENGTH = $CFG_VECTOR_LENGTH;
$DEC_CFG_AGC = $CFG_AGC;
$DEC_CFG_CMN = $CFG_CMN;
$DEC_CFG_VARNORM = $CFG_VARNORM;

$DEC_CFG_QMGR_DIR = "$DEC_CFG_BASE_DIR/qmanager";
$DEC_CFG_LOG_DIR = "$DEC_CFG_BASE_DIR/logdir";
$DEC_CFG_MODEL_DIR = "$CFG_MODEL_DIR";

#*******variables used in decoding of wave files *******
$DEC_CFG_DICTIONARY     = "$DEC_CFG_BASE_DIR/etc/$DEC_CFG_DB_NAME.dic";
$DEC_CFG_FILLERDICT     = "$DEC_CFG_BASE_DIR/etc/$DEC_CFG_DB_NAME.filler";
$DEC_CFG_LISTOFFILES    =
"$DEC_CFG_BASE_DIR/etc/${DEC_CFG_DB_NAME}_test.fileids";
$DEC_CFG_TRANSCRIPTFILE =
"$DEC_CFG_BASE_DIR/etc/${DEC_CFG_DB_NAME}_test.transcription";
$DEC_CFG_RESULT_DIR     = "$DEC_CFG_BASE_DIR/result";

# This variables, used by the decoder, have to be user defined, and
# may affect the decoder output

$DEC_CFG_LANGUAGEMODEL_DIR = "$DEC_CFG_BASE_DIR/etc";
$DEC_CFG_LANGUAGEMODEL  = "$DEC_CFG_LANGUAGEMODEL_DIR/an4itr.arpa.DMP";
$DEC_CFG_LANGUAGEWEIGHT = "8";
$DEC_CFG_BEAMWIDTH = "1e-700";
$DEC_CFG_WORDBEAM = "1e-080";

$DEC_CFG_ALIGN = "builtin";

#*******variables used in characterizing models*******

$DEC_CFG_HMM_TYPE = $CFG_HMM_TYPE;

if (($DEC_CFG_HMM_TYPE ne ".semi.") and ($DEC_CFG_HMM_TYPE ne ".cont.")) {
  die "Please choose one CFG_HMM_TYPE out of '.cont.' or '.semi.', " .
    "currently $DEC_CFG_HMM_TYPE\n";
}

# This comes directly from reading the code. The feature definitions
# aren're represented exactly by the same string in the trainer and
# the decoder. Therefore, we need to map between them.
%feature_type = (
      'c/1..L-1/,d/1..L-1/,c/0/d/0/dd/0/,dd/1..L-1/' => 's2_4x',
      'c/1..L-1/d/1..L-1/c/0/d/0/dd/0/dd/1..L-1/'    => 's3_1x39',
      'c/0..L-1/d/0..L-1/dd/0..L-1/'                 => '1s_c_d_dd',
      'c/0..L-1/d/0..L-1/'                           => 'cep_dcep',
      'c/0..L-1/'                                    => 'cep',
      'c/0..L-1/dd/0..L-1/'                          => 'INVALID',
      '4s_12c_24d_3p_12dd'                           => 's2_4x',
      '1s_12c_12d_3p_12dd'                           => 's3_1x39',
      's2_4x'                                 => 's2_4x',
      's3_1x39'                                     => 's3_1x39',
      '1s_c_d_dd'                                   => '1s_c_d_dd',
      '1s_c_d_ld_dd'                                => '1s_c_d_ld_dd',
      '1s_c_d'                                      => 'cep_dcep',
      '1s_c'                                        => 'cep',
      '1s_c_dd'                                     => 'INVALID',
      '1s_d'                                        => 'INVALID',
      '1s_dd'                                       => 'INVALID',
     );

$DEC_CFG_FEATURE = "INVALID"
    unless ((exists $feature_type{$CFG_FEATURE})
      and ($DEC_CFG_FEATURE = $feature_type{$CFG_FEATURE}));

if ($DEC_CFG_FEATURE eq "INVALID") {
  die "Feature type used for training, $CFG_FEATURE, cannot be used for
decoding.\n" .
    "Please use one of 1s_c_d_dd, 1s_c_d, 1s_c, s2_4x, s3_1x39,
1s_c_d_ld_dd\n";
}

$DEC_CFG_NPART = 1;   #  Define how many pieces to split decode in

$DEC_CFG_OKAY_COLOR = '00D000';
$DEC_CFG_WARNING_COLOR = '555500';
$DEC_CFG_ERROR_COLOR = 'DD0000';

return 1;
```