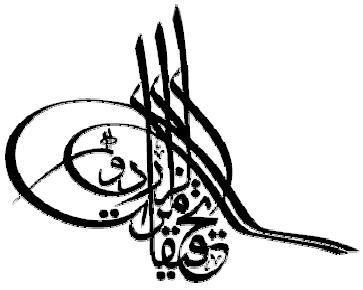
AUTOMATIC URDU DIACRITIZATION

Thesis

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Science

Abbas Raza ALI
July 2009



Department of Computer Science
National University of Computer and Emerging Sciences
&
Center for Research in Urdu Language Processing

Approved by

_____

Head of Department
Department of Computer Science
National University of Computer & Emerging Sciences

Approved by Committee Members

Advisor

_____

Dr. Sarmad Hussain
Professor
Department of Computer Science
National University of Computer & Emerging Sciences

Co-Advisor

_____

Dr. Mehreen Saeed
Assistant Professor
Department of Computer Science
National University of Computer & Emerging Sciences

Dedicated to my Parents

Acknowledgments

I am most grateful to Allah, who gave me thought, strength and determination to accomplish this task.

I am thankful to my advisor Dr. Sarmad Hussain and co-advisor Dr. Mehreen Saeed, for their supervision, guidance and encouragement throughout this work.

I am thankful to Ms. Madiha Ijaz who gave me this idea of research. She is always been very helpful during this work. I am also thankful to Mr. Aasim Ali and Mr. Amir Wali for their feedback and critical review of the dissertation.

Abbas Raza Ali

Table of Contents

List of Figures and Tables

1. Introduction

A diacritic, or a diacritical mark, is a small sign added to a letter in orthography to represent linguistic information. A letter which has been modified by a diacritic may be treated either as a new distinct letter, a modification of a letter or as a combination of two entities in orthography like اِن and اُن. This varies from language to language and, in some cases, from symbol to symbol within a single language. Diacritics are optional and usually not represented in Urdu orthography. Urdu speakers are able to restore the missing diacritics in the text based on the context and their knowledge of the grammar and lexicon. However, this could create problems for language learners, people with learning disabilities, and computational systems that require correct pronunciation.

Urdu is an Indo-Aryan language written in Arabic script. It is usually written without short vowels and other diacritic marks, often leading to potential ambiguity. While such ambiguity only rarely impedes proficient speakers, it is a source of confusion for beginning readers and people with learning disabilities. Diacritization is also problematic for computational systems, adding a level of ambiguity to both analysis and generation of text. For example, full vocalization is required for Text-To-Speech, Automatic Speech Recognition, and Machine Translation System to get unambiguous pronunciation of a word.

This thesis work presents analysis and implementation of automatic Urdu diacritization, by using statistical techniques and linguistic knowledge. The research work is divided into two main parts:

- to create Urdu tagged corpus, and lexicon; which includes orthographical, phonological, morphological, and syntactical information of a word.
- to build an appropriate hybrid models using the above data.

Section 2 will give a detailed analysis of Urdu language and overview of the previous relevant work on automatic diacritization will be discussed in Section 3. Section 4 will give the problem statement. Section 5 will discuss overall system architecture and algorithms used to implement the system. Section 6 provides a detailed discussion on data gathering and lexicon development; results by applying the algorithms (Section 5) on that data are recorded in Section 7. Detailed analysis after completion of the work and conclusion is given in Section 8 and 9 respectively.

2. Urdu Orthography

Urdu is written in Arabic script in Nastaliq style using an extended Arabic character set. The character set includes letters, diacritical marks, punctuation marks and special symbols [6]. It is a right-to-left script and shape assumed by the alphabet is context dependant [35]. Urdu support in Unicode is given in Arabic Script block. The details regarding alphabet, diacritics and special symbols have been provided ahead.

2.1. Alphabet

Urdu text comprises of the alphabet shown in Figure 1. Majority of the alphabets have been borrowed from Arabic and only a few have been borrowed from Persian and Sanskrit.

ا آ ب بھ پ پھ ت تھ ٹ ٹھ ث ج جھ چ چھ ح خ د دھ ڈ
ڈھ ذ ر رھ ڑ ڑھ ز ژ س ش ص ض ط ظ ع غ ف ق ک کھ
گ گھ ل لھ م مھ ن نھ ں و وھ ہ ء ی یھ ے

Table 2-1: Urdu Alphabet

2.2. Digits

Digits from 0 to 9 are represented in Urdu are shown in Figure 2.3.

۹ ۸ ۷ ۶ ۵ ۴ ۳ ۲ ۱ ۰

Table 2-2: Digits in Urdu

2.3. Special Symbols and Punctuation Marks

Special symbols and punctuation marks that may occur in Urdu text are shown in Figure 2.4. Their details can be found in Arabic script block in Unicode (http://www.unicode.org/charts/).

Table 2-3: Special symbols in Urdu

2.4. Diacritics

A diacritic is a mark placed above, through or below a letter, in order to indicate a sound different from that indicated by the letter without the diacritic [34].

Urdu has three short, eight long oral, seven long nasal vowels and various diphthongs. Long vowels are represented in orthography by combination of alif, wao and choti-yeh with diacritics zair, zabar and paish. Rest of the diacritical-marks is used as short vowels, adverbial markers and consonant doubling. They are also used to mark absence of vowel. Details of diacritical-marks and their usage are as follows [6]:

- Diacritics used for short vowels i.e. zair, zabar and paish merely change the sound value of the letter to which they are added (excluding alif, wao and yeh as when they are combined with these letters, they form long vowels e.g. بَل is /bəl/ while بَال is /bal/).
- Jazam represents absence of the vowel.
- Tashdeed represents germination i.e. doubling of consonants.
- The three short vowel diacritics i.e. zair, zabar and paish are doubled at the end of the word (do zabar, do zair, do paish) to indicate that consonant on which the vowel has been placed is followed by respective vowel and /n/; these vowels are called tanween.

Tanween represents grammar cases and it also serves as an adverbial marker in Arabic but in Urdu only do zabar is used and it acts as adverbial marker. Words containing tanween other than do zabar are Arabic words.

- Khari zabar indicates a long /a/ sound where alif is normally not written e.g. رحمٰن but it is also written as رحمان. But there are some words in which khari zabar cannot be replaced by Alif e.g. اعلٰی ,الٰہی etc. Again this phenomenon occurs in Arabic and it exists in Arabic loan words only.
- There are some other diacritical marks also that do not represent vowel e.g. zair-e-izafat (دلِ نادان /dɪl e na.dan/) and kasra-e-izafat (بازیچہ اطفال /ba.zi. ʧaɪ. əʈ.fal/) [6].

Diacritics described in Table 2-2 exist in Urdu text [36, 37].

Diacritical Marks	Description	Example	IPA
◌َ	Zabar (Fatah)	لَب	ləb
◌َ	Fatah Majhool	زَہر	zɛhɛr
◌ِ	Zair (Kasra)	دِل	ɖɪl
◌ِ	Kasra Majhool	اِہتِمام	eh.ʈe.mɑm
◌ُ	Paish (Zamma)	گُل	gʊl
◌ُ	Zamma Majhool	عُہدہ	oh.ɖɑ
◌ْ	Sakoon (Jazam)	سبزْ	səbz
◌ّ	Tashdeed (Shad)	ڈبّا	ɖəb.bɑ
◌ً	Tanween	فوراً	fɔ.rən
◌ٰ	Khari Zabar	عیسٰی	i.sɑ
◌ٰ	Elaamat-e-Ghunna	جنگ	ʤəŋ

Table 2-4: Diacritics in Urdu

2.5. Optical Vocalic Content

Urdu is normally written only with letters, diacritics being optional. However, the letters represent just the consonantal content of the string and in some cases (under-specified) vocalic content. The vocalic content may be optionally or completely specified by using diacritics with the letters [1]. Every word has a correct set of diacritics, however, it can be written with or without any diacritics at all, therefore, completely or partially omitting the diacritics of a word is permitted.

In certain cases, two different words (with different pronunciations) may have exactly the same form if the diacritics are removed, but even in that case writing words without diacritics is permitted. One such example is given below:

تَیر /ʈær/ (swim)

تِیر /ʈir/ (arrow)

However, there are exceptions to this general behavior; like certain words in Urdu require minimal diacritics without which they are considered incomplete and cannot be correctly read or pronounced. Some of these words are shown in Table 2-5.

Actual pronunciation	English tranlation	Urdu Translation with diacritics (correct)		Urdu translation without diacritics (incorrect)	
/ɑ.lɑ/	High quality	/ɑ.lɑ/	اعلیٰ	/ɑ.li/	اعلی
/ʈəq.ri.bən/	Almost	/ʈəq.ri.bən/	تقریباً	/ʈəq.ri.bɑ/	تقریبا

Table 2-5: Some Urdu words that require diacritics

3. Literature Review

This section provides brief discussion on previously held research on automatic diacritization. There are four major statistical approaches that are discussed in the literature for automatic diacritization.

3.1.1. Instance Based Learning Approach

Mihalcea [9] performed experimentation on four languages; Czech, Hungarian, Polish and Romanian for diacritization restoration. There are very few resources available for these languages, so no other knowledge sources are used except raw text. The data of those languages is collected over the internet, newspapers, and electronic literature. For training purpose corpus of 14,60,000 words for Czech, 17,20,000 words for Hungarian, 25,00,000 words for Polish, and 30,00,000 words for Romanian is used, out of which 50,000 examples are used for testing purpose. Instance based learning technique is used at letter-level for diacritics restoration. This technique simply stores the training examples and postpones its implication until a new instance is classified. In each iteration a new query instance is encountered its relationship to the previously stored examples. It is examined in order to assign a target function value for new instance [30]. The technique is very appropriate for the current scenario, because it requires no additional tagging information, which makes it language independent, particularly appealing for the languages for which there are few knowledge sources available. The maximum accuracy determined for all four languages is 98.17% and the detailed accuracies are given in Table 3-1.

Language	Training Data (words)	Baseline (%)	Overall (%)
Czech	14,60,000	80.44	97.83
Hungarian	17,20,000	75.32	97.04
Polish	25,00,000	87.18	99.02
Romanian	30,00,000	81.88	98.17

Table 3-1: Language wise detailed accuracies

Only ambiguous letters that contain multiple pronunciations are trained and a context window is defined for them. The above accuracy was achieved by setting the window size to 5 which means context of five letters on each side of the ambiguous letter.

3.1.2. Statistical and Knowledge based Approach

Vergyri [10] used two transcribed corpora; FBIS[1] consists of 2,40,000 words and LDC[2] consists of 1,60,000 words, for training and 48,000 words for testing purpose. Three techniques for Arabic diacritization are used; first combines acoustic, morphological and contextual information to predict the correct form, the second ignores contextual information, and the third is fully acoustics based. Most of the Arabic scripts can have a number of possible morphological interpretations. To identify all possible diacritization and assign probabilities to them; all possible diacritized variants for each word is generated, along with their morphological analyses. A standard HMM based statistical trigram tagging model is used in which undiacritized words and morphological tags are used as observed random variables. Correct morphological tag assignment was not known so unsupervised learning technique, Expectation Maximization, is used to iteratively train the probability distributions of the model. The best diacritics sequence is identified and their separate accuracies are measured for all three techniques, mentioned above, at word and character-level details are given in Table 3-2.

Knowledge Source	Word level (%)	Character level (%)
acoustic only	50.0	76.92
acoustic + morphological (tagger probability weight = 0)	72.7	86.76
acoustic + morphological + contextual (tagger probability weight = 1)	72.7	88.46
acoustic + morphological + contextual (tagger probability weight = 5)	72.7	88.06

Table 3-2: Results of Automatic diacritization of Arabic for Acoustic Modeling in Speech Recognition

[1] Foreign Broadcast Information Service (FBIS) is a collection of Arabic script transcribed radio news cast in Arabic.

[2] Linguistics Data Consortium (LDC) - consist of romanized transcript based telephonic conversation between native Arabic speakers.

Ananthakrishnan [11] used generative techniques for recovering vowels and other diacritics that are contextually appropriate. Their key focus is to develop techniques for automatic diacritization for speech recognition and NLP systems for Modern Standard Arabic (it is not concerned about dialectical variations). Simple N-gram based generative models integrated with more contextual and morphological information for predicting diacritics was used in their work. The dataset used by the above techniques is taken from Arabic Treebank[3] released by the LDC consists of 5,54,000 words. This data is divided into two sets - training set contains 5,41,000 words and test set of about 13,300 words. Their model of automatic diacritization consisted of both statistical and knowledge-based approaches. In statistical approach maximum likelihood based unigram technique is used as baseline mentioned in the following equation:

$$w_i^d = \arg\max_{w^d} P\left(w^d \mid w_i^u\right)$$

where w_i^d is the best diacritized form for the i^{th} word in the input undiacritized stream w_i^u. The word and character-level trigram language models are just the contextual expansion of the baseline model. Morphological analyzer and part-of-speech information is used as knowledge source which give them significant boost of 0.06 and 3.4% respectively. A maximum accuracy of 86.50% is recorded using trigram word-level model, tetra-gram character-level model, and part-of-speech knowledge source, details are given below.

Model	Accuracy (%)
Baseline	77.96
Word-level trigram	77.30
Character-level tetragram	74.80
Word trigram + character tetragram	80.21
Word trigram + morphological analyzer	80.27
Word trigram + part-of-speech	83.59
Word trigram + character tetragram + part-of-speech	86.50

Table 3-3: Results of statistical Arabic diacritization including knowledge-base sources

[3] Arabic Treebank released by LDC contains newswire text from AFP, Ummah, and An-Nahar.

Nelken [13] solved the problem of Arabic diacritization by using probabilistic finite-state transducers trained on the Arabic Treebank. The corpus is divided into training and test set with the ratio of 90% and 10%. Finite-state transducers are integrated with maximum likelihood based word and letter-level language models, and an extremely simple morphological model. The basic model consists of four transducers, mentioned in Figure 3-4.

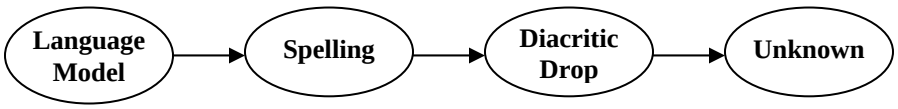


Figure 3-4: Basic model of Arabic diacritization using Finite-state transducers

Language model consists of a standard trigram of Arabic diacritized words. Weights of the model are learned from the training set. These weights are used to select the most probable word sequence that could have generated the undiacritized text. A spelling transducer is used to transduce a word into letters. Diacritic drop transducer is used for dropping vowels and other diacritics. It replaces all short vowels and syllabification marks with the empty string and also handles the multiple forms of the glottal stop. Unknown transducer is used to handle sparsity in data. During decoding phase, the letter sequence is fixed, and since it has no possible diacritization in the model. Using trigram word-level, clitic[4] concatenation and tetra-gram character-level model a maximum of 92.67% accuracy is achieved by the system.

Elshafei [15] trained the system based on domain knowledge e.g., sports, weather, local news, international news, business, economics, religion, etc. The training data consists of 33,629 diacritized words, composed of 260,774 characters. The test set consists of 50 randomly selected sentences from the entire Quran text; contains 995 words and 7,657 characters. Hidden Markov Model base approach is used to solve the problem of automatic generation of diacritical marks of Arabic text. Its training is consisted of word and letter level bigram and trigram technique. Following equation is showing the formulation of Bigram Arabic diacritization model:

[4] Clitic is a grammatically independent and phonologically dependent word, pronounced like an affix, but work at phrase level; like in English possessive *'s* is a clitic.

$$P(D|W) = P(d_1 . d_2 \cdots d_n | w_1 . w_2 \cdots w_n) = P(d_1 | w_1) \prod_{i=2}^{n} P(d_i | d_{i-1}; w_{i-1} . w_i)$$

After training, Viterbi algorithm is used to get optimal diacritics sequence of an unknown text. The bigram language model achieved 95.9% accuracy and improvements like preprocessing stage and trigrams for selected number of words is achieved about 97.5%. Errors of the system are divided into three classes. The first class errors are occurred due to inconsistent representation of tashkeel in the training set like لا، لاَ، لاَ. The second class errors are caused by a few articles and short words like إن، إنَّ. The third class of errors occurs in determining the boundary cases of words.

3.1.3. Expectation Maximization (EM) based Approach

Krichhoff [12] used the same corpora and also split training and test data same as [10]. The FBIS transcriptions corpus does not contain diacritics, so for automatic diacritization, all possible diacritized variants for each word is generated along with their morphological analyses. After that an unsupervised tagger is trained to assign probabilities to sequences of morphological tags. The trained tagger is used to assign probabilities to all possible diacritization sequences for a given utterance. It was used to train acoustic models from a different corpus to find the most likely diacritization. A standard trigram model is used but true morphological tag assignment was not known, only set of possible tags for each word were available during training. So that the probabilities and tag sequence models were updated iteratively using an unsupervised learning algorithm Expectation Maximization. The algorithm shows 95% accuracy on unknown Arabic text diacritization.

3.1.4. Maximum Entropy based Approach

Zitouni [14] used Maximum Entropy based approach for restoring diacritics in Arabic text. This approach is integrated with a wide array of lexical, segment[5] based and part-

[5] Segment is defined here as each prefix, stem or suffix.

speech tag features. The overall language model consists of statistical and features, implicitly learns the correlation between these types of diverse sources of information and the output diacritics. To train and test the above models, publically available LDC corpus is used. It consists of 340,281 words out which 288,000 words are used for training and 52,000 for testing purpose. Their algorithm is for formulated as a classification problem where each character is assigned a label (diacritical mark). Set of diacritical marks to predict or restore is represented as Y = $\{y_1, y_2 \ldots y_n\}$ and example space is represented by X has associated with a binary feature vector f (x) = ($f_1(x)$, $f_2(x) \ldots f_m(x)$). So the set of training examples together with their classifications is represented as $\{(x_1, y_1), (x_2, y_2) \ldots (x_k, y_k)\}$. A set of weights $\alpha_{i.j}{}^{i=1\cdots n}_{j=1\cdots m}$ are associated with each feature to maximize the likelihood of data during training phase.

$$P(y|x) = \frac{\prod_{j=1}^{m} \alpha_{i.j}^{f_j(x)}}{\sum_i \prod_j \alpha_{i.j}^{f_j(x)}}$$

The features used are divided into three categories: lexical, segment-based, and part-of-speech. By combining all theses features a maximum of 94.9% accuracy is achieved by the system.

4. Problem Statement

Urdu orthography does not provide full vocalization of the text and the readers are expected to infer short vowels themselves. Urdu speakers are able to accurately restore diacritics in a document, based on the context and their knowledge of the grammar and lexicon. Text without diacritics becomes a source of confusion for beginning readers and people with learning disabilities; and it becomes really difficult to infer correct pronunciation of a word computationally. Inferring the full form of a word is useful when developing Urdu speech and language processing tools e.g. text-to-speech system, automatic speech recognition, machine translation; since it is likely to reduce ambiguity in these tasks. This leads to the following problem statement;

Pronunciation of a word cannot be determined correctly in case it is either Out-of-Vocabulary or if it corresponds to multiple pronunciations e.g. سونا can be an adjective سُونا meaning "deserted" or verb سونا meaning "to sleep" or noun سونا meaning "Gold". So as a result analysis of the sentence is highly undermined.

Problem 1

Statistical approaches to natural language processing are currently well-established and they work very well, however, one of their disadvantages is that they require large amount of data on which the model is to be trained. Problem in this case is gathering a huge amount of Urdu corpus, and its diacritization. Table 4-1 is showing the statistics of diacritized datasets used for diacratics disambiguation of Arabic language.

Source	Corpus Size	Total
FBIS and LDC [10]	2,40,000 + 1,60,000	4,00,000
AFP, Ummah, and An-Nahar [11]	1,27,915 + 1,27,818 + 2,98,796	5,54,529
Penn Arabic Treebank [16]	2,88,000	2,88,000
Penn Arabic Treebank [17]	3,40,281	3,40,281

Table 4-1: Diacritized corpora used to train automatic diacritization system for Arabic

Problem 2

To build Urdu part-of-speech tagger that can provide useful information in determining correct pronunciation of a word. The tagger currently available is trained on 1,00,000 words and this number of words is insufficient to correctly POS tag raw text. To enhance accuracy of the POS tagger, training data is to be increased. POS tagger can disambiguate the correct pronunciation e.g. in the following sentence;

Raw text

پاکستان کے شمالی علاقے سربلند چوٹیوں سرسبز و شاداب وادیوں پہاڑوں کو چیرتی آبشاروں رومانی جھیلوں دیو قامت گلیشیرز بل کھاتے دریاؤں اور گھنے جنگلوں جیسے قدرتی حسن سے مالا مال ہیں۔[6]

Diacritized text

پَاکِستَان کے شُمَالِی عِلَاقے سَربُلَند چوٹِیوں سَرسَبز و شَادَاب وَادِیوں پَہاڑوں کو چِیرتی آبشَاروں رُومَانی جھِیلوں دیو قَامَت گلیشِیرز بَل کَھاتے دَرِیَاؤں اَور گھَنے جَنگلوں جَیسے قُدرَتِی حُسن سے مَالاَ مَال ہَیں۔

Ambiguous words are mentioned in Table 4-2 with their part-of-speech tags, which becomes the source of disambiguation in most of the cases.

Word	IPA	POS	Word	IPA	POS
جھیلوُں	/ʤʰe.l ũ/	Verb	جھِیلوں	/ʤʰi.lõ/	Noun
بِل	/bɪl/	Noun	بَل	/bəl/	Noun
حَسَن	/hə.sən/	Proper Noun	حُسن	/hʊsn/	Noun

Table 4-2: Some ambiguous words extracted from the above raw text and their disambiguation from diacritized text

[6] www.jang.com.pk

Table 4-3 clarifying problem 2 in more depth when statistical tagger is applied on an ambiguous sentence. The probability of first tag sequence is more than second and hence correct pronunciation will be بَچّے /bəʧ.ʧe/ (Noun) instead of بَچے /bə.ʧe/ (Verb).

Urdu Text	Tag	Bigram Probabilities		
		Word | Tag	Tag | Previous Tag	Total Probability
بچے کھیل رہے تھے	Noun Verb Aspect Tens	0.00075	0.033	2.48×10^{5}
بچے کھیل رہے تھے	Verb Verb Aspect Tense	0.00003	0.00099	2.98×10^{8}

Table 4-3: Probabilities are calculated from Urdu POS Tagger trained on 1,00,000 words

5. Methodology

This section will discuss the steps followed in the implementation of automatic Urdu diacritization. It is divided into two steps;

[1] Preparation of automatically diacritized and part-of-speech tagged corpus[7], with the help of lexicon[8] that has diacritized words along with the part-of-speech.

[2] Implementation of appropriate statistical language model based on the above data.

5.1. Diacritization Process Model

The System is divided into two main phases;

- in first phase Urdu lexicon is prepared manually, and Urdu corpus is prepared according to the domain knowledge to obtain the contextual information.
- in second phase different levels of statistical language models are prepared; lexicon and corpus are used for training and testing purpose.

Manually diacritized and part-of-speech tagged lexicons (detail is in Section 6), gathered from different sources, are used as input data. All lexicons are first pre-processed to make a single lexicon and then it is used to prepare a diacritized and part-of-speech tagged corpus, which is then used as word level contextual knowledge-source. After that HMM based bigram and trigram character level diacritization; a word level part-of-speech language model is prepared. When the system finds undiacritized text as an input, it first looks into pronunciation lexicon to get diacritized text and its part-of-speech. If the text is not found from the lexicon, it is passed to affixation module that diacritized the suffix, prefix and if possible root of every word in the text. This process is used to maximize consumption of knowledge-base resources. In case of out-of-vocabulary text, the system passes it to statistical module where trained probabilities are applied on that text to compute optimal sequence of diacritized text. The high level architecture of Urdu diacritization is also explained through Figure 5-1.

[7] The corpus was collected at Center for Research in Urdu Language Processing (CRULP)

[8] The lexicon was collected from multiple sources; it is manually POS tagged and diacritized at Center for Research in Urdu Language Processing (CRULP)

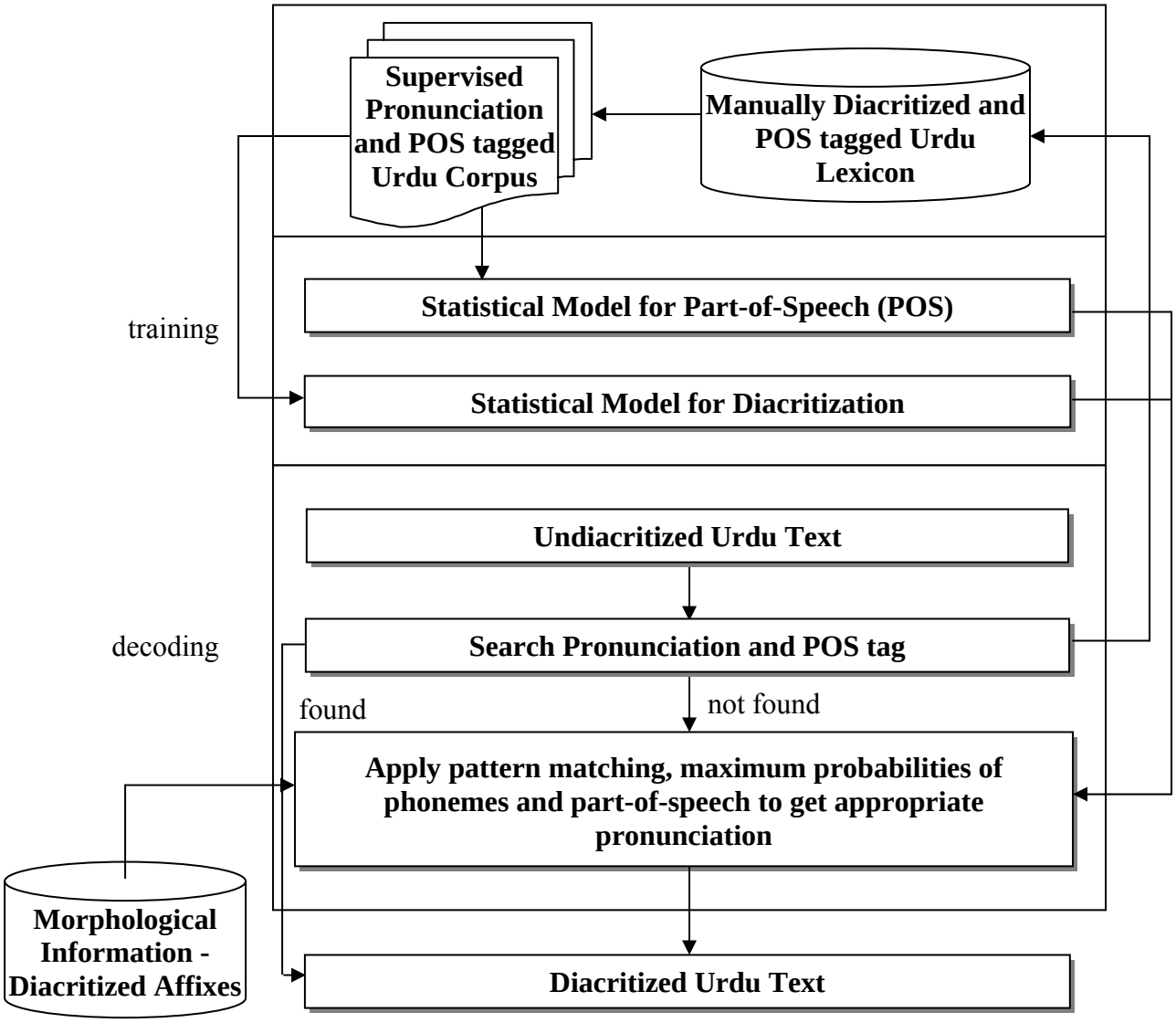


Figure 5-1: High-level architecture of automatic Urdu diacritization system

During the execution of the System the priorities are given to knowledge sources and statistical techniques, see Figure 5-2. First diacritics are removed from the input text then it is passed to normalizer to avoid duplicate version of the same character or word. After that the processed text is passed to part-of-speech tagger. The tagged data is then searched from lexicon in the form of <word, part-of-speech> and get diacritics version of the word. The words which are not found from lexicon are passed for affixation and the out-of-vocabulary words are passed to statistical diacritization module.

The morphological information and statistical language model will be applied on raw Urdu text based on its contextual information[9]. Following is the hierarchy of language model;

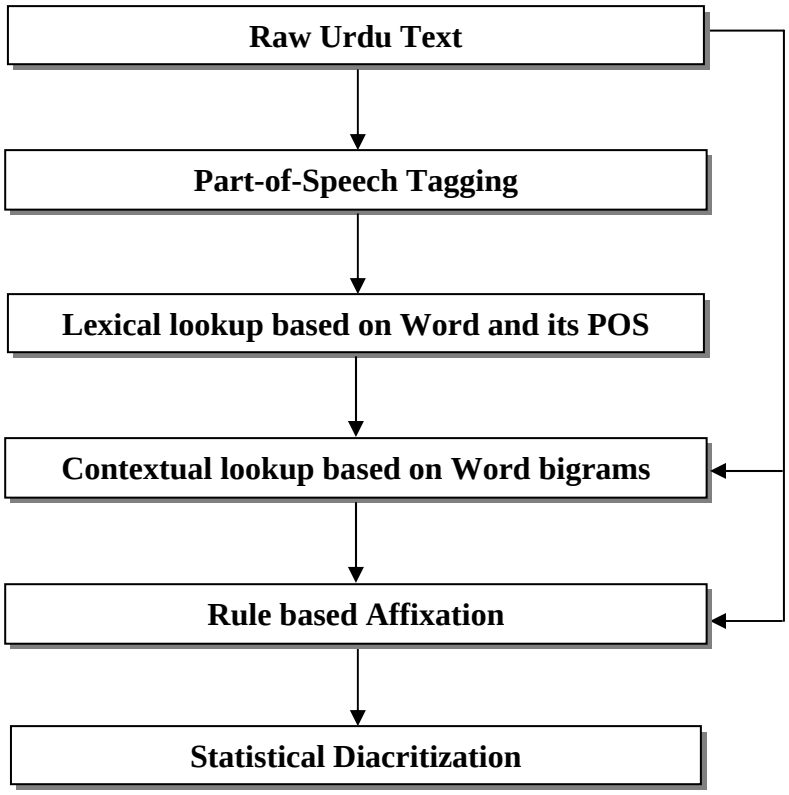


Figure 5-2: Hierarchy of knowledge sources and statistical model applicability

[9] Context information means how much contextual information is available for diacritization, like a single word, sentence, or paragraph.

5.2. Algorithms

Following are the algorithms that are used in the implementation phase of this research work.

5.2.1. Syllabification

Template matching technique is used for Urdu syllabification. In this technique syllabication can be done by matching template of the form $C_{0,1}VC^n$, starting from the end of the word towards its beginning [7]. Time complexity of the algorithm is O (W) where W is equal to length of word.

1. convert the entire input phoneme to consonant-vowel pairs
2. start from the end of the word
3. traverse backwards to find the next vowel
4. **repeat**
5. **if** there is a consonant preceding it
6. mark a syllable boundary before consonant
7. **else**
8. mark the syllable boundary before this vowel
9. **end if**
10. **until** the phonemic string is consumed completely

5.2.2. Diacritics Parameter Estimation

Hidden Markov Model is used to estimate the parameters of diacritization. It utilizes a diacritized and tagged corpus to estimate the frequency of the occurrences at character level.

Character-level Bigram Language Model

T_D = Diacritized Urdu lexicon

$V_D = dc_i\big|_1^N$ is diacritized vocabulary in T_D

$dc_k \, \varepsilon V_D$ = Diacritized characters of a word

F_D = Frequency of occurrence of each character in V_D

T_U = Undiacritized Urdu lexicon

$V_U = u_i|_1^N$ is undiacritized vocabulary in T_U

$u_k \, \varepsilon V_U$ Undiacritized character

F_U = Frequency of occurrence of each character in V_U

Let $V_D \rightarrow V_U$ be mapping from V_D to V_U

An undiacritized character sequence in a word;

$W = w_1 . w_2 \cdots w_N$

$w_t \, \varepsilon V_U$

$t = 1, 2, \cdots, N$

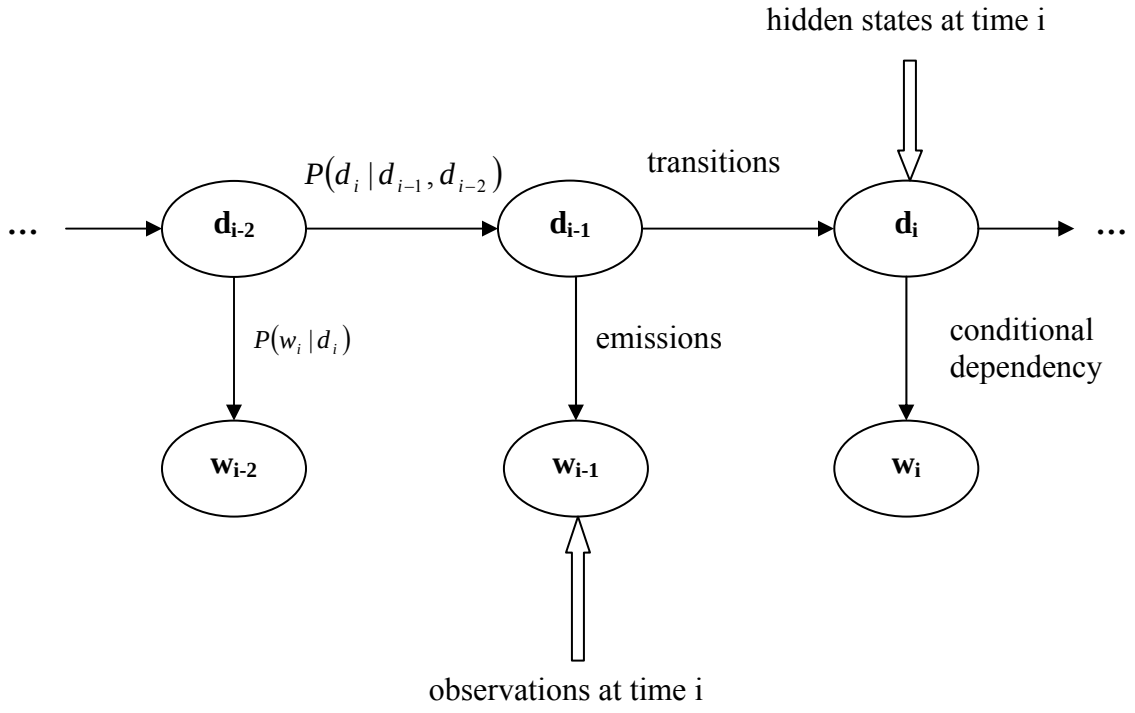


Figure 5-3: Architecture of Hidden Markov Model for Diacritization

To determine the most probable character sequence;

$D=d_1, d_2, \cdots, d_N$

$u_k \, \varepsilon V_U(j)$

$j=1,2,\cdots,N_d$

$w_t = u_k = v_u(k)$

$k=1,2,\cdots,N_u$

Diacritics sequence D may be chosen to maximize posterior probability. The best diacritized word sequence;

$$D=\arg\max_{D} P(D|W)$$

The conditional probability (using Bayes' Rule) can be written as;

$$P(D|W)=\frac{P(w_1 . w_2 \cdots w_n | d_1 . d_2 \cdots d_n).P(d_1 . d_2 \cdots d_n)}{P(w_1 . w_2 \cdots w_n)}$$

The probability of character sequence $P(w_1 . w_2 \cdots w_n)$ will be constant and can be ignored for maximization;

$$P(D|W)=P(w_1 . w_2 \cdots w_n | d_1 . d_2 \cdots d_n).P(d_1 . d_2 \cdots d_n)$$

$$P(D|W)=[P(w_1 | d_1 . d_2 \cdots d_n).P(w_2 | w_1; d_1 . d_2 \cdots d_n) \ldots P(w_n | w_1 . w_2 \cdots w_{n-1}; d_1 . d_2 \cdots d_n)].$$
$$[P(d_1).P(d_2 | d_1) \cdots P(d_n | d_1 . d_2 \cdots d_{n-1})]$$

To build special case of Trigram language model; each character is assumed to depend on its own diacritical mark and each diacritical mark is dependent only on its previous two diacritical marks;

$$P(D|W)=\left[\prod_{i=1}^{n} P(w_i | d_i)\right].\left[P(d_1).P(d_2 | d_1).\prod_{i=3}^{n} P(d_i | d_{i-1} . d_{i-2})\right]$$

Maximum likelihood estimation from relative frequencies will be used to estimate these probabilities;

$$P(w_i \mid d_i) = \frac{count(w_i, d_i)}{count(d_i)}$$

$$P(d_i \mid d_i \, . d_{i+1}) = \frac{count(d_i, d_{i-1}, d_{i-2})}{count(d_{i-1} \, . d_{i-2})}$$

Character-level Bigram Language Model

To build special case of Bigram language model; each character is assumed to depend on its own diacritical mark and each diacritical mark is dependent only on its previous diacritical marks;

$$P(D|W) = \left[\prod_{i=1}^{n} P(w_i \mid d_i) \right] . \left[P(d_1) . \prod_{i=2}^{n} P(d_i \mid d_{i-1}) \right]$$

Maximum likelihood estimation from relative frequencies will be used to estimate these probabilities;

$$P(w_i \mid d_i) = \frac{count(w_i, d_i)}{count(d_i)}$$

$$P(d_i \mid d_i \, . d_{i+1}) = \frac{count(d_i, d_{i-1})}{count(d_{i-1})}$$

5.2.3. Diacritics Parameter Optimization

Expectation Maximization (EM) algorithm is used to iteratively train the probability of word given diacritics P ($w_i \mid d_i$) and diacritical mark given previous and next diacritical mark P ($d_i \mid d_{i-1} \, d_{i-2}$) of the Hidden Markov Model. The general algorithm of Expectation Maximization is given below;

Initialization

1. **for each** Urdu orthography to pronunciation pair, assign equal probability combinations generated by language and pronunciation model.

2. **repeat**

Expectation

3. **for each** of the diacritical mark, count up instances of its different mappings from the observations on all pronunciation produced in section 5.2.2. Normalize the scores so that the mapping probabilities sum to 1.

Maximization

4. Recomputed the combination scores. Each combination is scored with the product of the scores of the symbol mappings it contains. Normalize the scores so that the mapping probabilities sum to 1.

5. **until** convergence

5.2.4. Computing Optimal Sequence of Diacritization

Viterbi algorithm will be used to compute the most probable diacritics sequence. The algorithm sweeps through all the diacritical mark possibilities for each word, computing the best sequence leading to each possibility. The idea that makes this algorithm efficient is that we only need to know the best sequences leading to the previous word because of the Markov assumption. Time complexity of the algorithm is O (W x D^2) where W is equal to length of the word and D is total number of diacritical marks. Figure 5-4 is showing an instance of computing the optimal diacritics sequence.

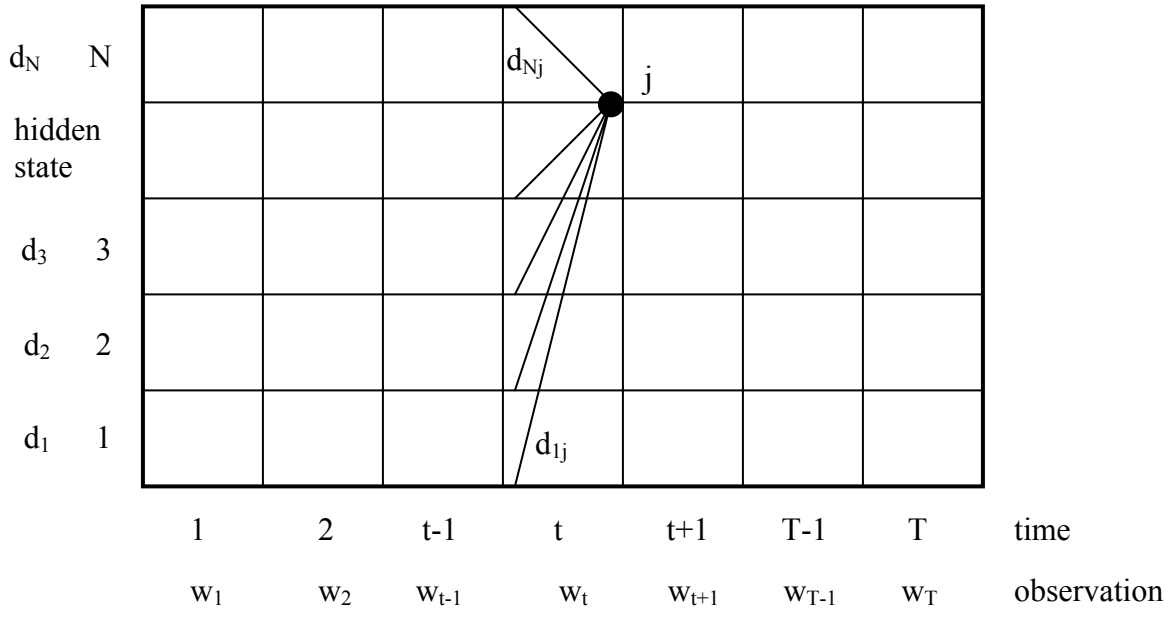


Figure 5-4: Computing the optimal sequence for diacritization

Initialization

1. **for each** diacritical mark j from 1 to D
2. $Score_{t, 0}$ = count (w_0, d_j) / count (d_j)
3. $Back\text{-}Pointer_{0, j} = 0$
4. **end for**

Induction

5. **for each** word i from 1 to W
6. **for each** diacritical mark j from 1 to D
7. $$Score_{i-1,\ j-1} = \max\left(Score_{i-1,\ j-1} \cdot \frac{count(w_i, d_i)}{count(d_i)} \cdot \frac{count(d_i, d_{i-1}, d_{i-2})}{count(d_{i-1}.d_{i-2})} \right)$$
8. $Back\text{-}Pointer_{i, j}$ = index that maximizes the score
9. **end for**
10. **end for**

Optimal Path

11. $Diacritic\text{-}Sequence_W$ = diacritical mark that maximizes $Score_{W, D}$
12. **for each** word i from W-1 to 1
13. $Sequence_i = Back\text{-}Pointer_{Sequence\ i,\ i+1}$
14. **end for**

5.2.5. Smoothing

Witten-Bell discounting technique will be used to assign some probability other than zero to unknown word given diacritics sequences in the data. It will be used to assign some probability other than zero to unknown sequence of word given diacritical mark.

- T is the number of types
- N is the number of tokens
- Z is the number of bigrams in the current data set that do not occur in the training data

1. **if** (count(w_i, d_i) = 0)

2. $$P(w_i|d_i)=\frac{T(d_i)}{Z(d_i).(N+T(d_i))}$$

3. **else**

4. $$P(w_i|d_i)=\frac{count(w_i,d_i)}{count(d_i)+T(d_i)}$$

5. **end if**

Deleted interpolation technique will be used to assign some probability other than zero to unknown sequence of diacritical mark given immediate previous and next diacritical mark sequences in the data. It combines different N-gram orders by linearly interpolating all three models in computation [28].

$$P(d_{i-1}, d_i, d_{i+1}) = \alpha_1 \cdot count(d_{i-1}, d_i, d_{i+1}) + \alpha_2 \cdot count(d_{i-1}, d_i) + \alpha_3 \cdot count(d_i, d_{i+1}) + \alpha_4 \cdot count(d_i)$$

α_1, α_2, α_3 and α_4 are constants and their sum must be equal to 1.

6. Data Preparation

Data of the same problem domain is the necessary part of statistical based systems; which is available in the form of corpus and lexicon contains system's domain knowledge information as well. It is observed from Section 3 - Literature Review that; morphological, syntactic, and phonological knowledge sources improve diacritization accuracy, so there are some manually prepared knowledge sources for Urdu will be used with the statistical techniques to improve the accuracy of overall system. Following are detail of these sources;

Data	Words
Corpus	2,50,000
Pronunciation and part-of-speech tagged Lexicon	1,65,000
Diacritized prefix including POS and type[10]	73
Diacritized suffix including POS and type	425

Table 6-1: Amount of data and knowledge sources

6.1. Lexicon Development

The diacritized and POS tagged lexicon is gathered from three different sources;

a. Text-to-speech lexicon[11], 85,000 word lexicon which provides information regarding diacritics, pronunciation and part-of-speech. The lexicon is using six part-of-speech tags namely Noun, Verb, Adjective, Adverb, Pronoun, and Harf. Format of Urdu pronunciation shown in Table 6-2.

Orthography	Diacritics	Pronunciation	Diacritics	POS
مزیدار	ZXXZXJ	مزےدار	ZXXZXJ	Adj_1
رسیوں	ZSRXJX	رسسیوں	ZJRXJX	Noun_1
زیرزمین	XJRZRXJ	زےرےزمین	XJRZRXJ	Noun_1

[10] Type means the affix is bound to be used with any other word or itself a word.
[11] The lexicon is developed at Center for Research in Urdu Language Processing (CRULP)

Table 6-2: Urdu Text-to-speech lexicon format

b. Online Urdu Dictionary, 81,000 words describing information regarding pronunciation, root word, etymology, and part-of-speech. The lexicon is using six part-of-speech tags namely Noun, Verb, Adjective, Adverb, Pronoun, and Harf. Format of Online Urdu Dictionary lexicon is shown in Table 6-2.

Orthography	IPA	Root Word	Etymology	POS
عارضی	ɑr.zi	ع رض	Arabic	Adjective
عدالت	ə.ḍɑ.ləṯ	ع د ل	Arabic	Noun
ڈرنا	ɖər.nɑ	-	Prakrit	Verb

Table 6-3: Online Urdu Dictionary format

c. Corpus based lexicon is of 50,000 common words and 53,000 proper nouns from other sources[12]; the lexicon describing pronunciation, part-of-speech, lemma[13], phonetic transcription and grammatical feature. It is using eleven part-of-speech tags including Noun, Verb, Adjective, Adverb, Pronoun, Numerals, Post Positions, Conjunction, Auxiliaries, Case Markers, and Harf. The pronunciation used in this lexicon is in SAMPA[14] not in IPA. A sample entry is given below;

```
<ENTRYGROUP orthography="مردوں">

    <ENTRY>
        <NOM class="common" case="oblique" number="plural"
        gender="masculine"/>
        <LEMMA>مرد</LEMMA>
        <PHONETIC>" m @ r - d_d o~</PHONETIC>
```

[12] Like Encyclopedia, Local Telephone Directory, Census Data etc.
[13] Lemma is a canonical form of a word.
[14] SAMPA stands for Speech Assessment Methods Phonetic Alphabets

```
        </ENTRY>
        <ENTRY>
            <NOM class="common" case="oblique" number="plural"
            gender="invariant"/>
            <LEMMA>مردہ</LEMMA>
            <PHONETIC" m U r - d_d o~</PHONETIC>
        </ENTRY>
    </ENTRYGROUP>
```

Table 6-4: Corpus based lexicon format

Using these three sources a synchronized lexicon is developed (Appendix D). Some information in the above lexica is not in the identical format like pronunciation, and detail of part-of-speech tags. The final lexicon consists of orthography, pronunciation, part-of-speech (Appendix C) and root language information of each word.

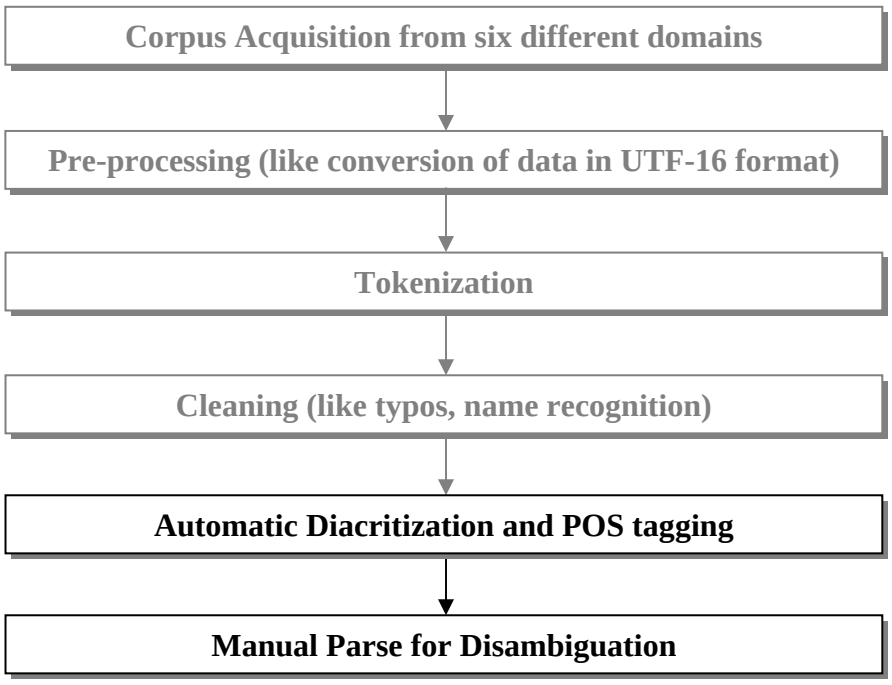
6.2. Corpus Development

6.2.1. Acquisition

The corpus acquisition and development for speech-to-speech done at CRULP for the creation of an Urdu lexicon needed for speech-to-speech translation. During this process various issues related to Urdu orthography were considered such as optional vocalic content, Unicode variations, name recognition, and spelling variation [2].

6.2.2. Automatic Diacritization and Part-of-Speech Tagging

In this thesis work some word-level language model and part-of speech tagging will demand contextual details as well. No diacritized and tagged corpus was available before this work, but a diacritized and tagged lexicon was available, through which a semi-supervised pronunciation corpus is prepared. Figure 6-5 outlines the procedure of building such a corpus.



Procedures used for the development of corpus

A corpus of 1,00,000 words is gathered from different sources for semi-supervised diacritization. Before diacritization the corpus is passed through a preprocessing phase like conversion of UTF-8 to UTF-16 to standardize the whole data, its cleaning … details are given in [2]. After that the cleaned corpus is first part-of-speech tagged through statistical tagger and then automatically diacritized from pronunciation lexicon by match word and tag of a word to increase the accuracy of diacritization. Then the diacritized corpus is manually parsed to remove errors and ambiguities which is 5% of the corpus.

7. Results

The following results have been extracted by using 10,143 diacritized and part-of-speech tagged words of the test corpus. The baseline accuracies are recorded by applying bigram and trigram techniques. After that syntactic, contextual, and morphological sources have been applied one by one and with combinations as well. Table 7-1 shows the detailed actuaries of the System.

Technique/Source	Bigram Accuracy (%)	Trigram Accuracy (%)
Baseline	81.13	84.07
POS based lexical lookup	90.86	91.83
Bigram lookup from corpus	89.06	90.75
Stemming	88.35	90.15
Bigram lookup from corpus + Stemming	91.91	92.77
POS based lexical lookup + Bigram lookup from corpus	93.86	94.35
POS based lexical lookup + Stemming	92.77	93.18
POS based lexical lookup + Bigram lookup from corpus + Stemming	**95.20**	**95.37**

Table 7-1: Accuracies of Urdu Diacritization

The candidate text has been first passed to a preprocessing phase. It consists of three modules normalization, un-diacritization, and tokenization of raw Urdu text. This preprocessed text is then passed to statistical diacritization module which is further categorized as Bigram and Trigram techniques. Baseline accuracies are calculated by using these two techniques separately on pronunciation and part-of-speech tagged lexicon data (Section 6). The baseline accuracies are then improved by applying different knowledge sources.

Two separate modules are used as knowledge sources which are part-of-speech tagger and stemmer. Part-of-speech tagger is trained on about 2,50,000 words corpus and after

applying HMM based bigram statistical technique 95.66% overall accuracy is achieved. A rule based stemmer is used to maximize the look-up which is more accurate then statistical technique. The stemmer module separates prefix, suffix and root of a word which is then lookup from a list of diacritize prefixes, suffixes and roots. The remaining part of the word; which is not found from the list passed to statistical module to complete the word's diacritization. The rule based stemmer module handles both inflectional and derivational morphology and shows about 91.2% accuracy.

After that, every combination of knowledge sources, mentioned above, are integrated with baseline system to get the maximum accuracy of overall system. From the results in Table 7-1 it is analyzed that the trigram technique is better than bigram but by adding knowledge-based sources, both techniques are generating almost equivalent results. Table 7-2 is showing the class-wise accuracies of the system.

Diacritical Mark	Accuracy (%)
Zair	69.42
Zabar	95.23
Paish	38.60
Jazam	93.44

Table 7-2: Class-wise Accuracies of Urdu Diacritization

8. Analysis

After performing some manual diacritization and experimentation on raw corpus, some assumptions were concluded that are as follow

Urdu diacritical marks are divided into three groups;

1. Zair, Zabar, Paish, and Jazam
2. Khari-zabar, Tashdeed, and Do-zabar
3. Hamza

The first group Zair, Zabar, Paish, and Jazam are catered in this work only, to predict words pronunciation statistically. These diacritical marks change the pronunciation of an Urdu word. Pronunciation rules are applied on the second group of diacritics, to eliminate them from training and test set as their probability of occurrence in the diacritized lexicon is very low, as mentioned in Table 8-1. The pronunciation rules are applied as follow

- Khari-zabar is usually comes with و and ی, and if this diacritical mark come with any of these letters then that letter its diacritical mark is replaced with ا letter. For example تقویٰ is modified as تقوا, and صلوٰۃ is modified as صلاۃ.
- Tashdeed usually comes with a pronunciation diacritical mark on single letter. Two copies of that letter are made. The first copy contains no diacritical mark, the pronunciation diacritical mark is attached with the second copy of letter and Tashdeed is removed. For example, پتّے is modified as پتتَے, and سَچّی is modified as سَچچِی.

- Do-zabar diacritical mark is usually occurred on letter ا, both letter and diacritical mark, are replaced by letter ن. For example, نسبتاً is modified as نسبتن and اتفاقاً is modified as اتفاقن.

Hamza is treated as a letter not diacritical mark.

Only 67,969 words contain partial diacritical mark in a corpus of 19.3 million words which is about 0.35%. In training corpus the number of times diacritical marks are occurred on a letter is shown in Table 8-1. In decoding phase it is analyzed that diacritical mark Jazam is appearing on first and last letter which is against the rules of Urdu language. It cannot occur in start, end and on the letters و, ا and ی when they are occurring as vowel; it usually comes in word medial position on the last letter of the syllable. From Table 8-1 it is observed that this is happened because of high frequency of Jazam in diacritized training data.

In the training data Urdu words' orthography and its pronunciation are usually not aligned (letter to diacritical mark alignment) which creates problem in statistical training process. One solution to that problem is statistically aligning of the word diacritical marks sequence through unsupervised learning technique. Through experiments it was found that the accuracy of word-diacritics statistical alignment is less than 72% which will decrease the overall systems' accuracy. For Example, word جراح is diacritized as ZSZXJ (جَرَّاحْ) where five diacritical marks are mapped on four letters of a word.

Three different diacritized training lexicons are used to train the System. All of them contain words' orthography, pronunciation and part-of-speech tags. Those lexicons are not synchronized like their pronunciation schemes, and part-of-speech tags are different. Some analysis regarding word sense disambiguation is also done at corpus and lexicon

level, in which 4.3% words with ambiguous pronunciation are found in diacritized corpus and 11.3% in pronunciation lexicon.

Diacritical Mark	Frequency	Percentage
◌َ	3,12,823	36.36
◌ِ	2,11,498	24.58
◌ُ	50,176	5.83
◌ً	2,84,604	33.13
◌ٰ	756	0.03
◌ّ	450	0.05
◌ٍ	335	0.02

Table 8-1: Occurrence of Diacritical Marks in the training set

From Table 7-2 and Table 8-1 it can be observed that the occurrence of Zer and Paish diacritical marks is very lower than the Zabar and Jazam. This huge difference between these two sets created problem for statistical module because in decoding phase Zabar and Jazam assigned more priority which generate more errors, and decrease overall system accuracy.

By comparing the results of this work with automatic Arabic diacritization work, where the maximum overall accuracy achieved is 97.50%. This system is still showing very prominent accuracies, because most of the automatic Arabic diacritization work used well known diacritized corpora and in Urdu language these types of recourses are not available. Corpus and lexicon preparation for training and testing of the system is the major part of that work. The accuracy mentioned in Table 7-1 can be improved by increasing training data set, minimize diacritization errors from the tagged data, and applying better statistical techniques like Support Vector Machine (SVM).

9. Conclusion

This work discusses in detail both the linguistic and computational aspects needed for the development of Automatic Diacritization System for Urdu language. Bigram and Trigram based Hidden Markov Model is applied over the training corpus of 250,000 words for part-of-Speech tagging, and 165,000 words for diacritization. The system showed maximum 95.37% accuracy while applying all knowledge-base sources along with statistical techniques. The overall accuracy can be increased by providing larger training data to the system, adding language specific rules and applying more sophisticated statistical techniques.

10. Future Work

This is the first effort on automatic Urdu diacritization and many improvements, in future, can be added in the system to improve its overall accuracy. Some improvements can be done after applying diacritized stemming on a word; if diacritization is applied separately on stem and its suffix then its pronunciation breaks. Other thing is that, sometimes diacritics depend on its next vowel, like Zair diacritical marks cannot occur before letter ا, و, and ے; and Paish diacritical mark cannot occur before letter ا, ی, and ے. This can be solved by applying these rules on the final diacritized words or applying these rules on training data before passing it to the learner.

Bibliography

[1] Butt, M. and King, T. H. "Urdu and the Parallel Grammar Project". *Proceedings of Workshop on Asian Language Resources and International Standardization*, Pages 39-45, 2002.

[2] Ijaz, M. and Hussain, S. "Corpus Based Urdu Lexicon Development". *Proceedings of Conference on Language Technology (CLT07)*, University of Peshawar, Pakistan, 2007.

[3] Hardie, A. "Automated Part-of-Speech Analysis of Urdu: Conceptual and Technical Issues". *Yadava, Y, Bhattarai, G, Lohani, RR, Prasain, B and Parajuli, K (eds.) Contemporary issues in Nepalese linguistics*. Katmandu, Linguistic Society of Nepal, 2005.

[4] Hussain, S. "Finite-State Morphological Analyzer for Urdu". MS Thesis. *Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences*, Lahore, Pakistan, 2004.

[5] Sajjad, H. "Statistical Part-of-Speech for Urdu". MS Thesis. *Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences*, Lahore, Pakistan, 2007.

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

[7] Hussain, S. "Phonological Processing for Urdu Text to Speech System". *Yadava, Y, Bhattarai, G, Lohani, RR, Prasain, B and Parajuli, K (eds.) Contemporary issues in Nepalese linguistics*. Katmandu, Linguistic Society of Nepal, 2005.

[8] Kominek, J. and Black, A. W. "Learning Pronunciation Dictionaries: Language Complexity and Word Selection Strategies". *Proceedings of the Human Language Technology Conference of the NAACL*, Pages 232-239. New York City, USA, 2006.

[9] Mihalcea, R. and Nastase, V. "Letter Level Learning for Language Independent Diacritics Restoration". *Proceedings of 6th Workshop on Computational Language Learning, CoNLL-2002*, 2002.

[10] Vergyri, D. and Kirchhoff, K. “Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition”. *Ali Farghaly and Karine Megerdoomian, editors, COLING-2004 Workshop on Computational Approaches to Arabic Script based Languages*, Pages 66–73. Geneva, Switzerland, 2004.

[11] Ananthakrishnan, S, Narayanan, S. and Bangalore, S. “Automatic Diacritization of Arabic Transcripts for Automatic Speech Recognition”. *Proceedings of ICON-05*, Kanpur, India, 2005.

[12] Kirchhoff, K. Vergyri, D. “Cross-Dialectal Data Sharing for Acoustic Modeling in Arabic Speech Recognition”. *Proceedings of Speech Communication*, Pages 37–51, May 2005.

[13] Nelken, R. and Shieber, S. M. “Arabic Diacritization using Weighted Finite-State Transducers”. *ACL-05 Workshop on Computational Approaches to Semitic Languages*, Pages 79–86, Michigan, 2005.

[14] Zitouni, I., Sorensen, J. S. and Sarikaya, R. “Maximum Entropy Based Restoration of Arabic Diacritics”. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Pages 577–584, Sydney, Australia, 2006.

[15] Elshafei, M., Al-Muhtaseb, H. and Alghamdi, M. “Statistical Methods for Automatic Diacritization of Arabic Text”. *The Saudi 18th National Computer Conference*, Pages 301-306, Riyadh, Saudi Arabia, 2006.

[16] Habash, N. and Rambow, O. “Arabic Diacritization through Full Morphological Tagging”. *Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL07)*, Rochester, New York, 2007.

[17] Mona, D., Ghoneim, M. and Habash, N. “Arabic Diacritization in the Context of Statistical Machine Translation”. *Proceedings of the Machine Translation Summit (MT-Summit)*, Copenhagen, Denmark, 2007.

[18] Elshafei, M., Al-Muhtaseb, H., Alghamdi, M. “Machine Generation of Arabic Diacritical Marks”. *Proceedings of the 2006 International Conference on Machine Learning; Models, Technologies, and Applications (MLMTA'06),* USA, June 2006.

[19] Davel, M., Barnard, E. “Extracting Pronunciation Rules for Phonemic Variants”. *Workshop on Multilingual Speech and Language Processing (MULTILING-2006)*, 2006.

[20] Rabiner, L. R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”. *Proceedings of the IEEE,* Pages 257–286, February 1989.

[21] Brill, E, “Transformation-based error-driven learning and Natural Language Processing: a case study in part-of-speech tagging”. *Proceedings of Computational Linguistics*, Pages 543-565, 1995.

[22] Goldsmith, J. “Unsupervised Learning of the Morphology of a Natural Language”. *Computational Linguistics*, Volume 27 No. 2, Pages 153-198, 2001.

[23] Tachbelie, M. Y., Menzel, W. “Sub-word Based Language Modeling for Amharic”. *Proceedings of the European Conference on Recent Advances in Natural Language Processing, RANLP-2007*, Borovets, Bulgaria, 2007.

[24] Clarkson, P. and Rosenfeld, R. “Statistical Language modeling using CMU-Cambridge Toolkit”. *Proceedings of EuroSpeech'97*, Pages 2707-2710, Rhodes, Greece, September 2007.

[25] Stolcke, A. “SRILM - An Extensible Language Modeling Toolkit”. *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pages 901-904, 2002.

[26] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M, Bertoldi, N, Cowan, B, Shen, W., Moran, C., Zens, R., Dyer, C., Bojar. O., Constantin, A., and Herbst, E. “MOSES - Open Source Toolkit for Statistical Machine Translation”. *Proceedings of the ACL 2007*, Pages 177–180, Prague, June 2007.

[27] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. “Bleu: A Method for Automatic Evaluation of Machine Translation”. *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pages 901–904, 2002.

[28] Jurafsky, D. and James, M. H. “Speech and Language Processing”. *Prentice Hall*, 2000.

[29] Alpaydin, E. “Introduction to Machine Learning”. *Prentice Hall*, 2006.

[30] Mitchell, T. M. “Machine Learning”. *McGraw-Hill*, 1997.

[31] Bokhari, R. and Pervez, S. “Syllabification and Re-Syllabification in Urdu”. *Akhbar-i-Urdu*, Pages 63-67, Lahore, Pakistan, 2003.

[32] Humayoun, M. "Urdu Morphology, Orthography and Lexicon Extraction". MS Thesis. *Chalmers University of Technology*, Sweden, 2006.

[33] E-Government Directorate (EGD) of Ministry of Information Technology (MoIT). “Online Urdu Dictionary”. *www.crulp.org/oud*, 2006.

[34] Wells, J. C. “Orthographic Diacritics and Multilingual Computing”. *In Proceedings of Language Problems and Language Planning*, 2001.

[35] Zia, K. “Standard Code Table for Urdu”. *In Proceedings of the 4th Symposium on Multilingual Information Processing, (MLIT-4)*, Yangon, Myanmar. CICC, Japan.

[36] Haq, M. A. “Qawaid-i-Urdu”. Lahore Academy, Pakistan.

[37] “Urdu Lughat, Taraqqi”. *Urdu Board Karachi*, Pakistan.

Appendix A - Urdu Phonemic Inventory

Consonants [6]

IPA	Letter	IPA	Letter	IPA	Letter	IPA	Letter	IPA	Letter	IPA	Letter
/b/	ب	/ḍ/	د	/s/	ص	/g/	گ	/bʰ/	بھ	/ɽʰ/	ڑھ
/P/	پ	/ɖ/	ڈ	/z/	ض	/l/	ل	/pʰ/	پھ	/kʰ/	کھ
/ṭ/	ت	/z/	ذ	/t/	ط	/m/	م	/ṭʰ/	تھ	/gʰ/	گھ
/ʈ/	ٹ	/r/	ر	/z/	ظ	/n/	ن	/ʈʰ/	ٹھ	/lʰ/	لھ
/s/	ث	/ɽ/	ڑ	/ʔ/	ع	/v/	و	/ʤʰ/	جھ	/mʰ/	مھ
/ʤ/	ج	/z/	ز	/ɣ/	غ	/h/	ہ	/ʧʰ/	چھ	/nʰ/	نھ
/ʧ/	چ	/ʒ/	ژ	/f/	ف	/ṭ/	ۃ	/dʰ/	دھ	/ŋʰ/	نگھ
/h/	ح	/s/	س	/q/	ق	/ʰ/	ھ	/ɖʰ/	ڈھ	-	
/x/	خ	/ʃ/	ش	/k/	ک	/ʔ/	ء	/ rʰ/	رھ		

IPA	Bilabial		Labio-Dental		Dental		Al-veolar		Retroflex		Post-Alveolar		Velar		Uvular| Glottal	
Plosive | Stops	p pʰ	b bʱ			ṭ ṭʰ	ḍ ḍʱ			ʈ ʈʰ	ɖ ɖʱ			k kʰ	g gʱ	q	ʔ
Nasal	m	Mʱ					n	nʱ					ŋ	ŋʱ		
Affricate											tʃ tʃʰ	dʒ dʒʱ				
Fricative			f	v			s	z			ʃ	ʒ	x	ɣ	h	
Trill							r	rʱ								
Lateral							l	lʱ								
Flap									ɾ	ɾʱ						
Approxi-mant											j					

Vowels [2]

IPA	Letter/Diacritical Mark	IPA	Letter/Diacritical Mark
/i/	ی	/ʊ/	ُ
/e/	ے	/ə/	َ، ء
/ɛ/	ـ	/ĩ/	ِیں
/æ/	َے	/ẽ/	یں
/u/	ُو	/æ̃/	َیں
/o/	و	/ũ/	ُوں
/ɔ/	َو	/õ/	وں
/ɑ/	ا، آ	/ɔ̃/	َوں
/ɪ/	ِ	/ɑ̃/	اں

Diacritics

IPA	Diacritics	Name	Conventions used in this work	Examples
/ə/	َ	Zabar	Z	رَنگ
/ɪ/	ِ	Zair	R	زِیارت
/ʊ/	ُ	Paish	P	سُکُون
/a/	ٰ	Khari Zabar	K	زکٰوۃ
/ən/	ً	Do Zabar	D	تقریباً
“	ّ	Tashdeed	S	بتّی
‘	ْ	Jezam	J	وٱلدیْن
-	-	Null Vowel	X	-

Appendix B - Affixes

Prefix

Prefix	POS	Type	Prefix	POS	Type	Prefix	POS	Type
اَز	NN|ADJ|ADV	Free	سَر	NN|ADJ	Free	آؤٹ	NN|ADJ	Free
اَن	HRF|VB|NN	Free	سه	NN|ADJ	Free	آٹو	NN	Free
با	NN|ADJ	Free	صاحِب	NN	Free	پرو	NN|ADJ	Free
باز	NN	Free	صَد	NN	Free	پرِی	NN|ADJ	Free
بَد	NN|ADJ	Free	غَم	NN	Free	پوسٹ	NN	Free
بَر	-	Bound	غَیر	NN|ADJ| ADV	Free	پَولی	NN|ADJ	Free
بَرائے	NN	Free	فَرو	NN	Free	ڈِس	NN|ADJ	Free
بِن	NN	Free	فَوق	ADJ	Free	سَب	ADJ	Free
بہر	-	Bound	گُل	NN	Free	سُپَر	NN	Free
بیش	NN|VB	Free	گُلُو	NN	Free	سُوڈو	-	Bound
پا	NN	Free	لا	NN|ADJ	Free	فور	NN	Free
بے	NN NN|ADJ| VB|ADV	Free	ما	NN|PRN	Free	مِس	NN|ADJ| VB	Free
پائے	NN	Free	مہا	NN|ADJ	Free	مَلٹِی	NN|ADJ	Free
پُر	NN|ADJ	Free	نا	NN|ADJ| ADV	Free	مِنِی	NN	Free
پَس	NN	Free	نیک	NN	Free	ہائپَر	NN|ADV	Free
پَن	NN|ADJ|HRF	Free	ہَشت	NN	Free	میل	NN	Free
پیش	ADJ|NN	Free	ہَفت	NN	Free	نان	NN|HRF	Free
تہہ	-	Bound	ہَم	NN|ADJ	Free	ہائپَر	NN|ADV	Free
خَرد	NN	Free	ہمہ	NN|ADJ	Free	…		

Suffix

Suffix	POS	Type	Suffix	POS	Type	Suffix	POS	Type
اَزَار	NN	Free	فَروش	NN	Free	گوئیوں	-	Bound
اَفرُوز	NN	Free	فِگار	NN|ADJ	Free	لَوحِیاں	-	Bound
اَفزَا	NN	Free	گَردانی	NN	Free	مِزاجِیوں	-	Bound
اَفزَائِی	-	Bound	گِرِفتَه	NN	Free	مَندِیاں	NN	Free
اَفشَاں	NN	Free	گِیر	NN	Free	نَاموں	NN	Free
اَندَاز	NN	Free	وَرڈ	NN	Free	ناکِیاں	-	Bound
اَندَازِی	NN	Free	لَوح	-	Bound	نَاموں	NN	Free
اَندَام	NN|ADJ	Free	سَگالِی	NN|ADJ	Free	نَفسِیوں	-	Bound
اَندُوز	-	Bound	آفرِینِیاں	NN	Free	نِگاروں	NN	Free
اَنگیز	-	Bound	بازِیاں	NN	Free	نَوِیسوں	-	Bound
باز	NN|ADJ	Free	بَندِیاں	NN	Free	وَرزِیوں	-	Bound
داروں	NN	Free	دانِیاں	NN	Free	طَراز	NN	Free
دَست	NN|ADJ	Free	دَرَاز	NN	Free	طَرازی	-	Bound
دِل	NN|ADJ	Free	ریزوں	NN	Free	بِین	NN	Free
دِہانی	NN	Free	ریزِیوں	-	Bound	تَراشِی	NN	Free
رَنگی	NN|NUM	Free	زادوں	-	Bound	گُساری	-	Bound
سَتان	NN|ADJ	Free	شِعارِی	NN	Free	گُزار	NN|ADJ	Free
سَرا	NN	Free	شَناسِی	-	Bound	بافِی	-	Bound
شَناس	NN	Free	فِگاروں	NN|ADJ	Free	بانِی	NN	Free
صورَت	NN|ADJ	Free	فَہمِیاں	-	Bound	نُماؤں	NN	Free
طَراز	NN	Free	کاروں	NN|ADJ	Free	…		

Appendix C - Part of Speech Tags

Noun	NN	قلم، لاہور، لڑکا
Verb	VB	ہے، بیٹھا، کھانا
Adjective	ADJ	خوبصورت، ایک، کافی
Adverb	ADV	آس پاس، یکا یک، سچ مچ
Pronoun	PRN	وہ، ہم، جو، کون
Harf	HRF	نے، کی، کے، مگر

Appendix D - Lexicon

Ortho-graphy	IPA	POS	Etym-ology	Ortho-graphy	IPA	POS	Etym-ology
عاجزانه	ɑ.ʤɪ.ˈzɑ.nɑ	ADJ	Arabic	باتھ	ˈbɑʈʰ	NN	English
عرفیت	ˈʊr.fɪ.jəʈ	NN	Arabic	فراڈ	fʊ.ˈrɑɖ	NN	English
تبدلی	ʈə.ˈbəɖ.ɖʊ.li	ADJ	Arabic	فرائی	fə.ˈrɑ.i	VB	English
تبصرہ	ˈʈəb.sɪ.rɑ	NN	Arabic	بابل	ˈbɑ.bɪl	NN	Hebrew
زعفران	zəf.ˈrɑn	NN	Arabic	جہنم	ʤə.ˈhən.nəm	NN	Hebrew
فخریہ	ˈfəx.rɪ.jɑ	ADJ	Arabic	یہودی	jʊ.ˈhu.ɖi	NN	Hebrew
غصیلی	ɣʊ.ˈsæ.li	ADJ	Arabic	فراٹا	fər.ˈrɑ.ʈɑ	NN	Local
کاپنا	ˈkɑp.nɑ	VB	Sanskrit	کتک	kɪ.ˈʈək	ADJ	Local
سیام	sɪ.ˈjɑm	ADJ	Sanskrit	بڑبڑانا	bʊɽ.bʊ.ˈɽɑ.nɑ	VB	Local
روندھا	ˈrun.ɖʰɑ	ADJ	Sanskrit	کن	ˈkən	NN	Turkish
چہچہانا	ʧəh.ʧə.ˈhɑ.nɑ	VB	Urdu	قزاقانہ	qəz.zɑ.ˈqɑ.nɑ	ADJ	Turkish
دتکارنا	ɖʊʈ.ˈkɑr.nɑ	VB	Urdu	چاقو	ˈʧɑ.qu	NN	Turkish
حولدار	hə.vəl.ˈɖar	NN	Urdu	یہیں	jə.ˈhĩ	ADV	Hindi
بٹائی	bə.ˈʈɑ.i	NN	Urdu	تاڑنا	ˈʈɑɽ.nɑ	VB	Hindi
بٹھل	ˈbəʈʰ.ʈʰəl	NN	Pashto	تاشی	ˈʈɑ.ʃi	ADJ	Hindi
گندی	ˈgʊn.ɖi	NN	Pashto	کوٹھی	ˈko.ʈʰi	NN	Prakrit
تازیانہ	ʈɑ.zɪ.ˈjɑ.nɑ	NN	Persian	رت رہے	ˈrʈ rə.ˈhe	ADV	Prakrit
بادرفتار	ˈbɑɖ rəf.ˈʈar	ADJ	Persian	رتاوا	rə.ˈʈɑ.vɑ	NN	Prakrit
بادکش	ˈbɑɖ.kəʃ	NN	Persian	اصفہانی	ɪs.fə.ˈhɑ.ni	ADJ	Pahlavi
فراخ	fə.ˈrɑx	ADJ	Persian	…			