Phonological Processing for Urdu Text to Speech System

Sarmad Hussain

Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, B Block, Faisal Town, Lahore, Pakistan
Sarmad.hussain@nu.edu.pk
http://www.crulp.org

Abstract. Determining and modeling phonological phenomena is necessary to generate speech from textual input. These phenomena include letter to sound conversion, syllabification, sound change, stress assignment and intonation assignment. This paper presents work on Urdu phonological processes and provides algorithms to convert textual input into phonologically annotated output, required for Urdu text-to-speech system. Current paper builds on earlier work on letter to sound conversion rules and adds details of syllabification, sound change rules and stress assignment algorithm. Intonation assignment module is still under investigation and is not discussed in this paper.

1 Introduction

A text-to-speech (TTS) system for any language would input "raw" text and output corresponding speech. This conversion can be divided into three steps[1]: Natural Language Processing, Text Parameterization and Speech Synthesis. The first stage converts text into normalized textually annotated phones. The second stage converts the annotations produced in first stage into numeric parameters, e.g. phone duration and source frequency targets. The final stage uses these parameters to generate digital speech. This is illustrated by a high-level schematic[2] shown in Figure 1 below. The Natural Language Processor (NLP) can be sub-divided into a Text Pre-Processor, which normalizes the input text (e.g. converts alpha-numeric-string input into alpha-string output), and a Phonological Processor (PP), which converts normalized text to annotated phone string. This paper focuses on the Phonological Processor for Urdu TTS system, and other modules are not discussed any further in this paper.

[1] Dutoit divides the process into two stages: Natural Language Processing and Speech Synthesis [8].

[2] The schematic is based on an Urdu TTS system being developed by Urdu Localization Project at Center for Research in Urdu Language Processing (www.crulp.org); see [12] for details.

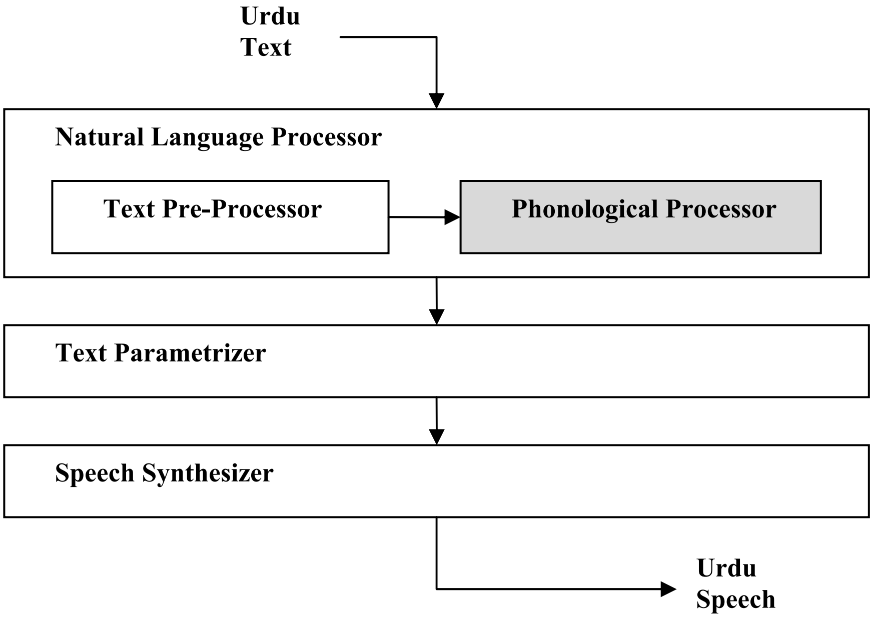

Fig. 1. High-level schematic for a TTS system

The paper is divided into multiple sections. The first part explains the requirements of a PP module for Urdu TTS system. The second section presents relevant phonological analysis and associated algorithms to realize the PP module for Urdu.

2 Phonological Processor (PP) Module

The Phonological Processor takes normalized textual input and outputs phonologically annotated text. The PP module is further divided into sub-processes. The first in this series of processes is Urdu letter-to-sound (LTS) conversion. This module takes in normalized Urdu text string and converts it into its phonemic equivalent. The phonemic output is marked with syllable boundaries by a Syllabification module. Syllabification is required to condition Urdu sound-change rules to convert the phonemic string generated by LTS module into corresponding phone representation for eventual output. This phone string is re-syllabified, in case of application of epenthesis or deletion rules. In the next module, the resulting syllabified phone string is marked for stress. Stress markers are essential in realizing the durational changes due to lexical stress [4] and for placement of accents for intonation. In the final module, this string is annotated with intonation pattern. This process is shown in Figure 2 below.

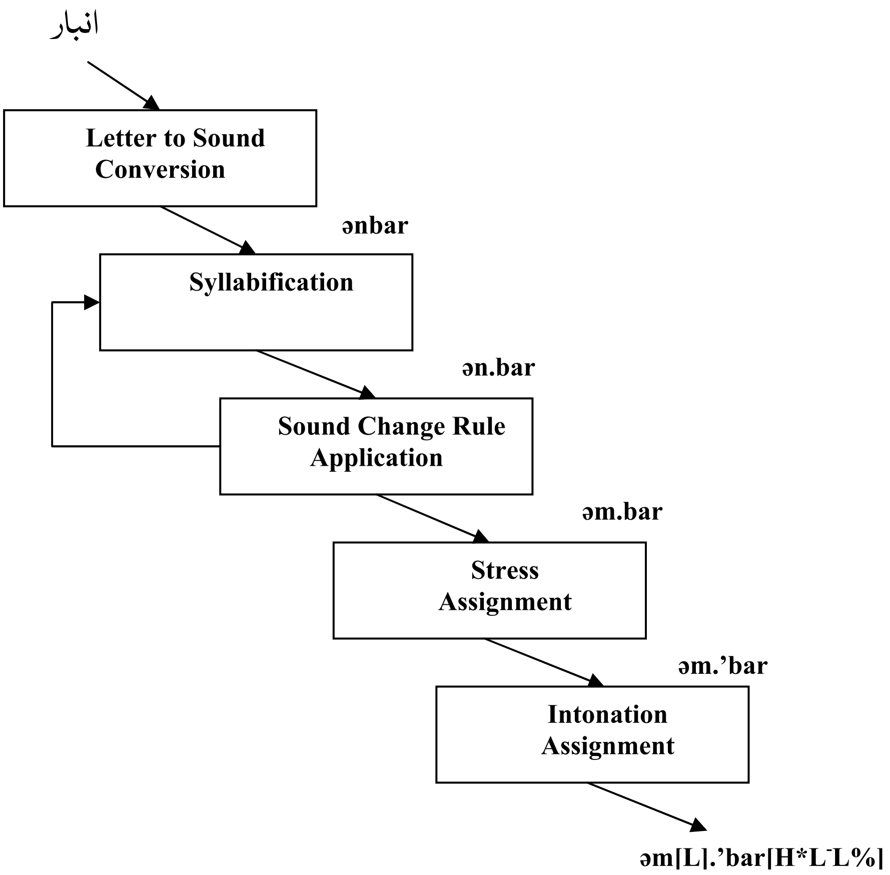

Fig. 2. Modular representation of Phonological Processor pipeline for Urdu

2.1 Letter-to-Sound Conversion

Urdu has a fairly regular mapping between its graphemic and phonemic representations. The details of Urdu graphemic and phonemic inventories and mapping between them are discussed in detail elsewhere [1]. However, this algorithm assumes that the vowel marks or diacritics are fully specified in Urdu input text. Writing these diacritics is optional in Urdu writing system and they are normally left out. Thus, this component works in conjunction with an Urdu lexicon which contains these diacritics for each word. For words with exceptional pronunciation (e.g. چھ “six” is pronounced [tʃ^{h}e] instead of [tʃ^{h}ə]), the diacritics are not encoded and the pronunciation is directly retrieved from the lexicon. For words not in the lexicon, e.g. proper nouns, a heuristic module assigns the diacritics before these letter-to-sound rules are applied. This module currently has some basic rules, e.g. Urdu cannot have *zer* (or *Kasra*, Unicode U+0650) before an *Alef* (U+0627). Work is under progress to investigate effective statistical measures to further enhance the Pronunciation Guesser module. LTS process in illustrated in Figure 3 below.

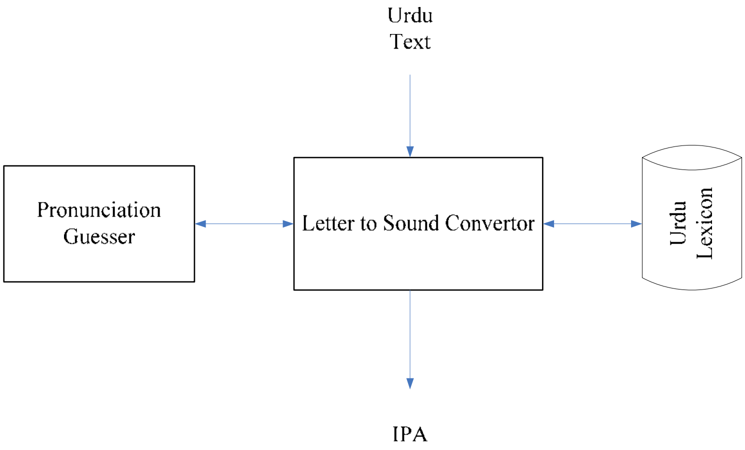

Fig. 3. Letter-to-Sound conversion process

The letter-to-sound rules are realized through a finite state transducer (FST), which inputs Urdu text and outputs corresponding IPA. As an example, Figure 4 below shows a part of the transducer which processes Urdu *Do-Zabar* (U+064B), which only comes with *Alef* (U+0627) in Urdu. The string اً produces the phoneme string /ən/

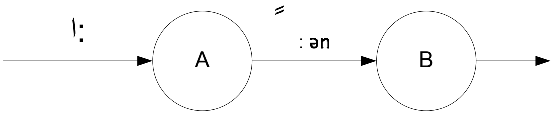

Fig. 4. Letter-to-Sound transducer (partial view for processing اً)

The algorithm for LTS conversion is as follows:

i) for input string, search the lexicon of exceptions

ii) if found, return the completely annotated string with exceptional pronunciation

iii) else search the regular lexicon

iv) if found, return the diacritic string

 a. convert to phonemic string using LTS rules [1]

v) else, call Pronunciation Guesser and get a guess on the diacritic string

 a. convert to phonemic string using LTS rules

2.2 Syllabification

Syllabification is a well studied phonological phenomenon (e.g. see [2], [3]). Syllables are formed by high-sonority nuclei with falling sonority going outward towards the edges of the syllable from this nucleus (onset and coda), as generalized in Sonority Sequencing Principle[3] (SSP). In addition to SSP, Maximal Onset Principle (MOP) states that given a consonant in the middle of two syllables and the possibility that it may be taken up in coda of a previous syllable or onset of the next syllable (i.e. it does not violate SSP in either case), languages prefer to maximize the onset by taking this consonant as part of the onset of the next syllable. Syllabification for languages has been done by either projecting nuclei and then using SSP and MOP in conjunction to incorporate the other phonological material or by using syllable Consonant-Vowel (CV) templates and fitting them from right to left (or left to right), e.g. see [3].

Work has been done on determining the syllabification mechanism for Urdu [4], [5], [6]. Both template matching [4], [6] and Nucleus projection based [5] techniques have been proposed. It is also argued in [5] that MOP does not hold for Urdu as it does not take complex onsets (i.e. more than a single consonant in the onset position), but may take complex codas and extra syllabic material at word final position. This constraint can be effectively exploited to syllabify a phonemic string of Urdu. Syllabication can be done by matching $C_{0,1}VC^*$[4] template from the end of the word towards its beginning, as illustrated by examples in Figure 5. The template matching starts from the end of the word. Intermediate states show intermediate steps in the syllabification process.

پاکستان	“Pakistan”:	pakɪst̪an → pakɪs.t̪an → pa.kɪs.t̪an → pa.kɪs.t̪an
تحقیقات	“Research”:	t̪əhkikat̪ → t̪əhki.kat̪ → t̪əh.ki.kat̪ → t̪əh.ki.kat̪
کائنات	“universe”:	kaenat̪ → kae.nat̪ → ka.e.nat̪ → ka.e.nat̪

Fig. 5. Syllabification of Urdu phonemic string by applying $C_{0,1}VC^*$ template from word-end (intermediate syllabified strings shown by underlined text)

The examples also show that intervocalic consonants are taken up as onsets. Where there is an inter-vocalic consonant cluster, its last consonant is taken as onset and rest are taken up as coda consonants. However, there may be onset-less syllables if there is no intervocalic consonantal material available. This behavior remains unchanged for short and long vowels (see [1] for vocalic inventory of Urdu).

[3] SSP may be “violated” at word edges, where extra-syllabic material may also attach. See [2] for a more detailed discussion.

[4] $C_{0,1}$ means zero or one consonant, C^* means zero or more consonants, V means a single (short or long) vowel.

Thus the algorithm for syllabification is as follows:

i) if the string is not exceptional (see Section 2.3), convert the input phoneme string to C(onsonant)-V(owel) string
ii) start from the end of the word
iii) traverse backwards to find the next V
iv) if there is a C preceding it, mark a syllable boundary before C
v) else mark the syllable boundary before this V
vi) repeat from step (iii) until the phonemic string is consumed completely

2.3 Sound Change Rules

Like other languages, Urdu also displays a variety of sound change rules due to coarticulation, giving a modified surface or phonetic form to represent the underlying phonemic string. Phonemic form is evident by the orthographic representation of words in many cases (e.g. see [1]).

Some of these rules are listed in Figure 6. Linear (and not auto-segmental) rule-format is given.

Bilabial assimilation	n → [+bilabial] / _ [+bilabial,-nasal]
Velar assimilation	n → [+velar] / _ [+stop,+velar,-nasal]
Nasal assimilation	V [+long] → [+nasal] / _ [+nasal]
/h/ deletion and vowel lengthening	V [+short] h → [+long]#
/h/ deletion	h → ø / V [long] _#

Fig. 6. Some sound change rules of Urdu represented in conventional linear format. Capitalized 'V' indicates a vowel and '.' indicates a syllable boundary

The algorithm followed for phonetic string generations is as follows:

i) if the string is not exceptional (see Section 2.2), starting from first phoneme
ii) for each phoneme in the input, run all the sound change rules in the order given
iii) repeat from step (ii) until the input is consumed

2.4 Stress Assignment

Urdu stress is sensitive to syllable weight. This weight can be represented by moraic count of each syllable [7]. Long vowels are "heavier" than short vowels. Thus, long vowels are bi-moriac and short vowels are mono-moraic in Urdu. In addition, each coda consonant has a weight equivalent to a single mora [4], [9]. Table 1 below shows the moraic count of various syllable templates of Urdu. Syllables can be mono-moraic (light), bi-moraic (heavy) and tri-moraic (super-heavy, e.g. closed syllables with long vowels).

Table 1: Moraic count of various Urdu syllable templates (VV represents a long vowel, V represents a short vowel, C represents a consonant)

Urdu Syllable Template	Moraic Count
CV	1
CVV	2
CVC	2
CVVC	3
V	1
VV	2
VC	2
VVC	3

Table 2 below shows some words of Urdu with stress assignments. These stresses are marked after consulting [10] and native speakers[5] (latter preferred if variation was observed between the two sources).

Table 2: Urdu words and their stress assignments

Urdu Word	English Translation	IPA Transcription
بیٹا	son	‘be.ta
تقدیر	fate	‘‘t̪ək.‘dir
عبرانی	jewish	‘‘ɪb. ‘ra.ni
پیشانی	forehead	‘‘pe.‘ʃa.ni
جھلملی	shiny	‘dʒʰɪl.mɪ.li
اصطلاحات	terminology	‘‘ɪs.t̪ə.‘la.‘‘hat̪
قسطنطنیا	Constantinople	‘‘kʊs.t̪ʊn.‘t̪ʊn.ja

Earlier analysis based on [10] (e.g. [4] and [9]) had a single stress marked for each word. However, feedback from the speakers indicates multiple stresses on each word as marked in Table 2 above[6]. The stresses marked show the preferred stresses in case multiple may be possible.

Analysis shows that heavy and super-heavy syllables may take primary or secondary stress. Primary stress is assigned to the first bi-moraic or tri-moraic syllables from the end of the word. Light syllables do not take stress. However, final syllables

[5] First-language-Urdu speakers growing up in Lahore, Pakistan.

[6] This may be because of differences in styles of Urdu spoken in different regions. Variation in stress is also noted in [11], though a single stress per word is marked.

do not take stress even if they are heavy, indicating that the final mora is extrametrical[7] [4].

Each heavy syllable causes perception of stress, causing variability in stress assignment. However, majority of speakers prefer assigning stress to the final stressed heavy syllable (after making adjustments to syllable weight for extrametricality). Secondary stresses are assigned to the other heavy syllables preceding the final heavy syllable. If there are more than one non-light syllables preceding the last non-light syllable, alternate is de-stressed (to avoid stressing too many syllables). Some words deviate from these rules. However, closer analysis shows that these words have morpheme boundaries, with each morpheme bringing its own stresses and following the stress assignment mechanism summarized above (except that the syllable final mora in non-final morphemes is not extrametrical), e.g. ‘‘ɪs.t̪ə.‘la+‘‘hat̪ (‘+’ indicates a morpheme boundary).

Figure 7 below shows the metrical structure. Each bi-moraic and tri-moraic syllable projects at foot level. Any light syllables are incorporated within a foot with the non-light syllable on its right.

There can be stress variation within minimal pairs to indicate part-of-speech (POS) changes, similar to English, e.g. ‘per.fect vs. per.‘fect. For Urdu, some of these words include الٹا, گرا, پکڑا. There is no direct way of differentiating between them without tagging it for POS using a tagger or parser.

[7] Another argument which supports extrametricaliy of word final mora is the fact that Urdu does not license light syllables in word final position. This is perhaps because extrametricality would render such syllables weightless.

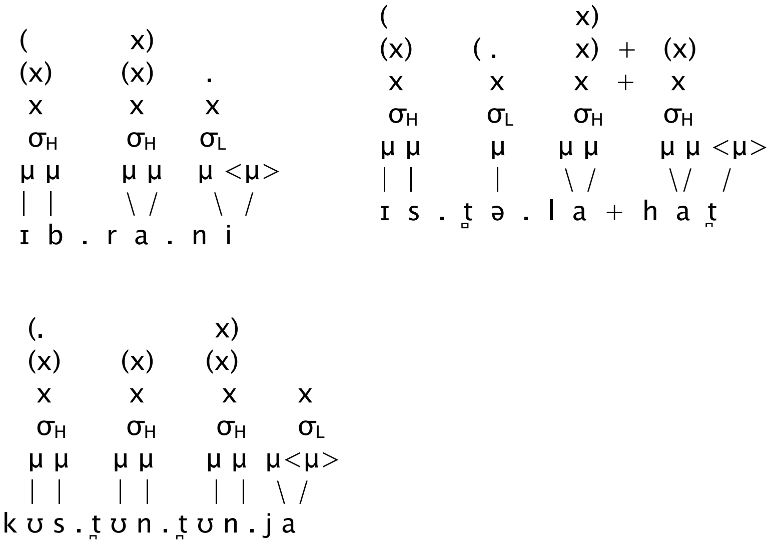

Fig. 7. Metrical structure for words of Urdu. H and L indicate "Heavy" and "Light" syllables respectively.

Stress is assigned using the following algorithm (excluding stress variation based on POS, as discussed above):

```
i)     for each syllable in the input phone string
         a. calculate the mora count
ii)    for the last syllable decrement mora count for extrametricality
iii)   identify all the morpheme boundaries (would need a morphological parser or
       stemmer for this step)
iv)    for each morpheme
         a. starting from the final syllable moving backwards, mark the first non-
            light syllable with stress
         b. if more syllables are left, repeat from step (iv. a)
v)     for the root morpheme
         a. mark the final stressed syllable with primary stress
```

A rule-based system is implemented using the algorithm described above to mark the stresses. The current algorithm is based on stresses marked by [10]. However, it is currently being extended to mark multiple stresses, as indicated. The current algorithm also needs to be extended to include a morphological parser to determine morpheme boundaries and use POS information to make any changes in stress assignment within minimal pairs.

3 Discussion and Conclusions

The paper discusses the Phonological Processor. Most of the work presented has been realized within the system under development. However, work is still under progress for realizing intonation assignment, and to guess pronunciation of words not in the lexicon. In addition, a single stress is currently being marked in the system, which corresponds to the primary stress in most words (except for words with multiple morphemes, where non-root morpheme also contains a non-light syllable). This algorithm also needs to be extended to include morphological and syntactic analysis.

More work also needs to be done on the determining the reasons and predict the variation in stress placement by speakers. As indicated, though majority of speakers prefer certain stress patterns, all indicate that there are alternative patterns which also do not sound un-natural. Acoustic dimensions of these variations also need to be investigated beyond what has been done earlier [4].

Current work is being integrated with other components in Figure 1, including the Text Parameterizer and Speech Synthesizer. Progress in this context will be presented in future.

Acknowledgements

This work has been partially supported by the grant for Urdu Localization Project by E-Government Directorate of Ministry of IT, Govt of Pakistan. Author is also thankful to the research staff and students for their valuable comments.

References

1. 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)
2. Goldsmith, J. A.: Autosegmental & Metrical Phonology. Basil Blackwell, Cambridge MA (1990)
3. Kenstowicz, M.: Phonology in Generative Grammar. Blackwell, Cambridge, USA (1994)
4. Hussain, S.: Phonetics Correlates of Lexical Stress in Urdu. Unpublished PhD Dissertation, Northwestern University (1997)
5. Akram, B.: Analysis of Urdu Syllabification using Maximal Onset Principle and Sonority Sequencing Principle. Akhbar-e-Urdu. National Language Authority, Pakistan (April-May 2002)
6. Nazar, N.: Syllable Templates of Urdu Language. Akhbar-e-Urdu. National Language Authority, Pakistan (April-May 2002)
7. Hayes, B.: Metrical Stress Theory, Principles and Case Studies. University of Chicago Press, Chicago (1995)
8. Dutoit, T.: An Introduction to Text-to-Speech Synthesis. Kluwer Academic Publishers, Dordrecht, The Netherlands (1997)
9. Coleman, J., Dirsksen, A., Hussain, S. and Waals, J.: Multilingual Phonological Analysis and Speech Synthesis. Proceedings of Second Meeting of the Association of Computational Linguistics: Special Interest Group in Phonology, Assoc. of Comp. Ling., P. O. Box 6090, Soerset, NJ 08875 (1996)
10. Standard Twentieth Century Dictionary: Urdu to English. Educational Publishing House, New Dehli, India
11. Kachru, Y.: Hindi-Urdu. In: Comrie, B. (ed.): The Major Languages of South Asia, The Middle East and Africa. Routledge, London (1990) 53 – 72
12. Hussain, S.: Urdu Localization Project. Proceedings of Workshop on “Computational Approaches to Arabic Script-based Languages,” COLING 2004, Geneva, Switzerland (2004)