Text Processing for Urdu TTS System

Rida Hijab Basit (rida.hijab@kics.edu.pk) & Sarmad Hussain (sarmad.hussain@kics.edu.pk)
Center for Language Engineering, Al-Khawarizimi Institute of Computer Science
University of Engineering & Technology, Lahore, Pakistan

Abstract

Natural Language Processing plays an important role in any Text to Speech (TTS) system. The raw text given as input to TTS may consist of numbers, dates, time, acronyms or symbols. NLP processes the raw text and converts it in the form that can be used by TTS to generate its corresponding speech. NLP consists of three parts; "Text Processing", "Text Annotation" and "Phonological Annotation". This paper enhances earlier work and details the text processing in NLP from the perspective of Urdu and also reports the results given by NLP.

Introduction

Text to Speech (TTS) system for any language takes a sequence of words as input & converts it into speech. TTS system can be divided into three parts: Natural Language Processing, Text Parameterization & Speech Generation.

The raw text given as input to TTS system can be of any form. It may consist of numbers, time, dates, symbols and any miscellaneous characters. Therefore, before converting it into speech it must be converted to some form that can be spoken by the TTS system. For this purpose, raw text first undergoes the process of Natural Language Processing.

Natural Language Processing normalizes input text and converts it to its corresponding phonetic transcription.

NLP Architecture

Natural Language Processing can be divided into three categories—Text Processing, Text Annotation & Phonological Annotation. Figure 1 shows high level diagram for NLP. The shaded portion has been explained in detail in this paper.

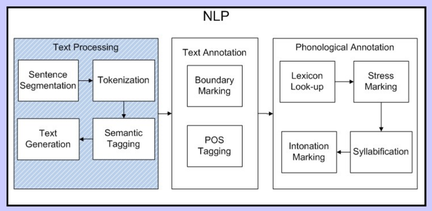

Figure 1: NLP Architecture

Text Processing Module

Text Processing module takes raw input from the user & converts it into normalized text. It mainly consists of sentence segmentation, conversion to Urdu Zabta Takhti (UZT), tokenization, semantic tagging & text generation. Figure 2. shows text processing module.

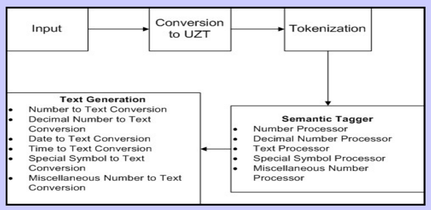

Figure 2: Text Processing Module

The input text undergoes sentence segmentation. These segmented sentences are then converted to 8-bit UZT format for internal processing. Tokenization module then tokenizes these sentences into words or equivalent units & sends them to the semantic tagger. Semantic tagger analyzes multiple tokens and where necessary labels them as numbers, time, date text or some other category. Each tagged token is then passed to text generation module which converts the token into its corresponding text string based on its label.

Sentence Segmentation

This module breaks the input string into sentences - if it encounters a full stop, question mark, line feed or carriage return character. In addition, if a sentence is very long, it is segmented at a hard limit of 400 characters (slightly shorter strings are truncated to adjust word boundaries).

Conversion to Urdu Zabta Takhti (UZT)

The segmented sentence, initially in Unicode format, is then converted to UZT format using a conversion map. Conversion is limited to the characters which are available in UZT. Other characters are ignored at this time, except ASCII digits, which are mapped onto Urdu digits.

Tokenization

The tokenization module separates words in the input string according to space and the punctuation, including ( ) ' " ! : / - ؛ ،etc. It also contains a few rules for specific cases, for example, to separate a number and text joined together like the string 12 بجےinto separate tokens; to identify a decimal number between digits as separately from an end of sentence marker.

Semantic Tagger

Semantic tagger analyzes the tokens—tagging them as date, time, numbers (whole numbers, fractional numbers & decimal numbers), text, special symbols and miscellaneous strings. This is done by considering each token (separated in the tokenization phase) in its context to see if they can be grouped into larger semantic forms. For example, 22/10/2010 should be converted to "بائیس اکتوبر سن دو ہزار دس" and not as "بائیس سلیش دس سلیش دو ہزار اور دس". This is done by first making "22/10/2010" strings as a single semantic unit and tagging it as a date, but "22/10" will be tagged as a fractional number.

It consists of various sub-modules depending upon the type of input text.

Text Generation

This module converts the tagged tokens sent from semantic tagger into their corresponding text equivalents. It has been divided into several sub-modules:

- Number to Text Converter
- Date to Text Converter
- Time to Text Converter
- Special Symbol to Text Converter
- Miscellaneous to Text Converter

Each sub-module covers different types of their respective domains.

Number to Text Converter

It deals with whole numbers, decimal numbers and fractional numbers. The numbers can have both English and Urdu digits. Some examples of different formats of these three types have been shown in Table 1. It covers 4 formats for whole numbers, 2 for decimal numbers & 2 for fractional numbers.

Input	Output	IPA Transcription of Output	English Translation
100000	ایک لاکھ	eːk laːkʰ	One lac
۱۰۰,۰۰۰	ایک لاکھ	eːk laːkʰ	One lac
12324	بارہ ہزار تین سو چوبیس	baːraː həzaːr ʈiːn sɔ ʧɔːbiːs	Twelve thousand three hundred twenty four
۸۹۳۳	آٹھ ہزار نو سو تینتیس	aːtʰ həzaːr nɔ sɔ ʈeːnʈiːs	Eight thousand nine hundred thirty three
213901	دو لاکھ تیرہ ہزار نو سو ایک	ɖoː laːkʰ ʈeːraː həzaːr nɔ sɔ eːk	Two lac thirteen thousand nine hundred one
21345320	دو کروڑ تیرہ لاکھ پینتالیس ہزار تین سو بیس	ɖoː kəroːɽ ʈeːraː laːkʰ pæːnʈaːliːs həzaːr ʈiːn sɔ biːs	Two crore thirteen lac forty five thousand three hundred twenty
12.02	بارہ اعشاریہ دو	baːraː iːʃaːrjaː ɖoː	Twelve point two
۱۳.۱۱	تیرہ اعشاریہ ایک ایک	ʈeːraː iːʃaːrjaː eːk eːk	Thirteen point one one
۶/۱۰	چھ بٹا دس	ʧʰ bəʈaː ɖɔs	Six by ten
1/3	ایک تہائی	eːk ʈʰaːiː	One third

Table 1: Number to Text Conversion in NLP

Date to Text Converter

Date to Text Converter handles three different types of dates, which are:

- D(D)-M(M)-Y(Y) & D(D)/M(M)/Y(Y)
- D(D)-Month-Text-Y(Y) & Month-Text D(D), Y(Y)
- YY(YY)

Here, D(D) is the date, M(M) is the month in numbers and Y(Y), a year. Month-Text is the month already in Urdu string. Years before 2000 have different Urdu string as output, as compared to the years after 2000. For example, 1994 will have a different output string as compared to 2002.

During text conversion, it also keeps track of symbols like Arabic Sanah, Hijri & Eeswin. Each type has different number of formats. Some examples have been shown in Table 2.

Input	Output	IPA Transcription of Output	English Translation
12-2-2000	بارہ فروری سن دو ہزار	baːraː fərvəriː sən ɖoː həzaːr	Twelve February year two thousand
۱۲/۲/۲۰۰۰	بارہ فروری سن دو ہزار	baːraː fərvəriː sən ɖoː həzaːr	Twelve February year two thousand
فروری 12 ،2000ء	بارہ فروری سن دو ہزار عیسوی	baːraː fərvəriː sən ɖoː həzaːr ʔiːsəviː	Twelve February year two thousand A.D.
فروری ۱۲ ،۲۰۰۰	فروری بارہ سن دو ہزار	fərvəriː baːraː sən ɖoː həzaːr	February twelve, year two thousand
1992ھ	سن انیس سو بانوے ہجری	sən uniːs sɔ baːnVeː hɪʤriː	Year nineteen ninety two A.H.
ء۲۰۰۱	سن دو ہزار ایک عیسوی	sən ɖoː həzaːr eːk ʔiːsəviː	Year two thousand one A.D.

Table 2: Example of different formats of date

Time to Text Converter

Any time entered is converted to text through this module. Time can have both English or Urdu digits. It covers 6 different formats of time. Table 3 shows some conversion of time entered in NLP.

Input	Output	IPA Transcription of Output	English Translation
5:00	پانچ	paːnʧ	Five
۴:۴۰	چار بج کر چالیس منٹ	ʧaːr bəʤ kər ʧaːliːs mɪnʈ	Four Forty

Table 3: Time examples in NLP

Special Symbol to Text Converter

56 different symbols are handled by this converter. Table 4 shows some examples.

Input	Output	IPA Transcription of Output	English Translation
@	ایٹ	æːʈ	At (Symbol)
ز	ز	zeː	Urdu character
ؓ	رضی اللہ عنہ	rəziː əllaːhu ʔənhuː	Arabic Symbol
$	ڈالر	ɖaːlər	Dollar
سنہ	سن	sən	Year
ع	ع	ʔiːn	Urdu character

Table 4: Special Symbols in NLP

Miscellaneous String to Text Converter

Miscellaneous strings consist of (:), (/) or (-) in some combination with text or numbers. All these are converted to their respective texts. Some examples are shown in Table 5.

Input	Output	IPA Transcription of Output	English Translation
جون:۲۰۰۱-۲۰	جون دو ہزار ایک ڈیش بیس	ʤuːn ɖoː həzaːr eːk dæːʃ biːs	June two thousand one dash twenty
12-12	بارہ ڈیش بارہ	baːraː dæːʃ baːraː	Twelve dash twelve
۱۲-۱۲/	بارہ ڈیش بارہ ڈیش بارہ سلیش	baːraː dæːʃ baːraː dæːʃ baːraː səleːʃ	Twelve dash twelve dash twelve slash
جون-جون/	سلیش جون ڈیش جون	səleːʃ ʤuːn dæːʃ ʤuːn	Slash june dash june
جون-2001:20	جون ڈیش دو ہزار ایک بیس	ʤuːn dæːʃ həzaːr eːk biːs	June dash two thousand one twenty

Table 5: Miscellaneous Strings in NLP

Results

	Total Tokens	Correctly Identified	%age Accuracy
Whole Numbers	271	267	99%
Dates	92	84	91%
Miscellaneous Strings	40	37	93%
Symbols	62	31	50%
Time	5	5	100%
Decimal Numbers	16	16	100%

Table 6: Accuracies of different modules

Conclusion

The paper has discussed various steps in detail, needed for converting raw text string into normalized Urdu string. Raw input may consist of numbers, date, time or symbols that must be normalized using text processing module before sending it to TTS system for speech generation. The overall accuracy for text processing module is **90.5%**, which is a quite acceptable number. However, this is a work in progress and some future goals are yet to be achieved. Future goals include refining NLP output and handling more formats for each sub-module depending upon the requirements.