Language Processing Chapter: Dzongkha

Chungku, Dechen Chhoeden, Jurmey Rabgay, Pema Choejey, Sithar Norbu, Tenzin Dendhup, Uden Sherpa, Yeshey Pelden,Yumkee Lhamo

Department of Information Technology, Ministry of Information and Communications, Bhutan.

1. Introduction (Script/language)

1.1. Script background

The script used in writing Dzongkha is identical to the Tibetan script and is known as 'Uchen' script. There are two basic forms of writing in Dzongkha: the Bhutanese formal longhand 'jotshum' and the Bhutanese cursive longhand 'joyig' [1]. While the formal longhand jotshum shares similarity with Tibetan orthography, joyig is distinctively unique to Dzongkha. In contrast to jotshum, joyig is not only cursive but also has a special abbreviated way of writing a letter or sequence of letters at the end of a syllable.

i) Character Set [2]

Dzongkha character set has thirty consonants and five vowels, including the /a/ vowel sound, which is inherent in all consonants. For e.g, ཀ is pronounced with the inherent vowel /a/ as /ka/.

ཀ ཁ ག ང ཅ ཆ ཇ ཉ ཏ ཐ ད ན པ ཕ བ མ ཙ ཚ ཛ ཝ ཞ ཟ འ ཡ
ར ལ ཤ ས ཧ ཨ

Figure 1.1. Dzongkha Consonants

We also have additional consonants in Dzongkha which were used to write loan words borrowed from Sanskrit. Nowadays they are normally used to write foreign loan words.

ཪ ཫ ཬ ཀྵ ཥ ཧྐ

Figure 1.2.

The figures below depicts the vowels in Dzongkha.

a) Vowels of Dzongkha

b) Example of the vowels combined with the consonant 'ཀ'

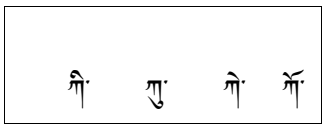


Figure 1.3.

Dzongkha has a set digits as follows:

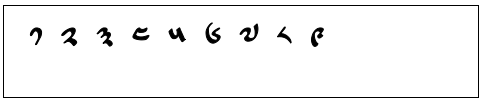


Figure 1.4. Dzongkha digits

Special characters used as punctuation marks:

Table 1.1.

Name	**Example**
Tsheg “ ་ ”	ངའི་ཚལ་ཡོད་དོ། (The Tsheg mark appears after every syllable)
Shed “ ། ”	ངའིཚལ་ཡོད་དོ། (Here it marks the end of a sentence.) It is used to mark the end of an expression (statement, sentence,

	paragraph).
Nyi Shed (Two Sheds) " ‖ "	Is usually used to end all lines in poems and proses. Also used in religious and story books.

1.2. Language history

Dzongkha is the official and the national language of Bhutan. The word "Dzongkha" means the language (Kha) spoken in the Dzongs, which are fortress like structures built around the 17th century to serve as both the religious and the administrative centers. In the western region, where it is widely spoken, it is formally known as "Ngalong Kha".

Dzongkha is a South-Bodish language derived from the Tibeto-Burman sub-family, see Figure. 1.5, from the Sino-Tibetan group of languages [3].

Sino-Tibetan

Tibeto-Burman

Himalayish

Tibeto-Kanauri

Tibetic

Tibetan

Southern

Dzongkha

Figure 1.5. Language Family Tree

Therefore, Dzongkha is linguistically related to the Classical Tibetan or Choeke (language of Buddhism) [4]. Choeke was used to write all the literary works of Bhutan. It is still used in religious texts and in other scholarly works.

As far as the early 1960's, Choeke was used as the language of learning and liturgy. It was only in the 1960's that Dzongkha replaced Choeke as the language of education in schools. Therefore the vocabulary of spoken and written Dzongkha is heavily influenced by Choeke. Its influence is similar to that of Latin over Roman languages [5].

Of 18 other languages, Dzongkha was chosen as the official language, because it was the only Tibeto-Burman language native to Bhutan with a written form [4]. It was made as the official language of Bhutan with a singular objective to unify the country and to initiate people's participation in the country's affairs, thereby making it the lingua franca of the nation.

1.3 Speaking population and geographical distribution of language

Dzongkha is spoken as the native tongue in eight western districts of Bhutan - Thimphu, Paro, Haa, Punakha, Gasa, Wangduephodrang, Dagana and Chukha. It is spoken as the first language by approximately 130,000 people and as the second language by about 470,000 people [1].

1.3. Any other relevant information

There are 19 different languages spoken across the country which either belong to the Central Bodish or the East Bodish forms [4] of the Tibeto-Burman family of languages. However, Dzongkha is the only language that has native literary tradition in Bhutan [4]. Lhotshamkha or Nepali which is the prominent language of the southern region is the only Indo-Aryan language. Apart from Dzongkha, the languages Tshanglakha and Lhotshamkha are widely spoken in the country owing to the large distribution of its native speakers in the eastern and southern regions respectively.

The medium of education in Bhutan is mainly English. Therefore most Bhutanese who have been educated in schools know how to speak and write in English.

2. Corpus

2.1. Description

The size of the Dzongkha corpus is approximately 400,000 words (600,000 syllables). The texts have been collected from different domains such as arts, religion, government and sports. It covers different genres like Reportage (political, cultural, sports), Press Editorials, Press Reviews, Travel Adventures, Biographies, etc. The texts collected so far have been sourced mainly from dictionaries, printed books, the print and broadcast media, and from relevant websites. All texts for the corpus were stored in xml format.

2.1.1. Sources

To design a balanced corpus for Dzongkha, text of different genres like newspaper articles, samples from traditional books, novels, dictionaries, scientific, sports, medical and social science are collected. So as to make it representative of every linguistic phenomena of

Dzongkha, in our collections of 600,000 syllables (approximately 400,000 words) following specific sources of 3 domains were included.

1) Arts
   a) Department of Education, b) Institute of Language and culture Studies, c) Bhutan Times (Media), d) Poems and articles by Jurmey Rabgay, e) Dzongkha Development commission.

2) Social Science
   a) Bhutan Times and b) Bhutan Broadcasting Service (Media)

3) World Affairs.
   Bhutan Broadcasting Service (Media)

2.2. Encoding conversion

For text to be usable by computer, it must include some kind of mark-up and annotations. The mark-up that is to be introduced into Dzongkha Text Corpus indicates explicitly a wide range of important information, including:

- The boundary and part of speech of each word.
- The sentence structure identified by POS (part of speech) Tagging Annotation.
- Paragraphs, sections, headings and similar features in written text.
- Meta-textual information about the source or encoding of individual texts

These textual features, and others, are all encoded in standardized way, to help ensure that the corpus will be usable no matter what the local computational set-up may be.

i) Criteria for selection of Encoding text

1. How long the code written will be compatible.
2. How convenient is it to extend the work (speech and parallel corpora).
3. Complexity in mark-ups in special cases.
4. All the conversion techniques based on XML/SGML mark-up languages.

ii) Conversion techniques and tools

The conversion stage involves transforming the corpus into the toolkit's own internal format, and checking for subtle errors that can only be picked up with a complete parser. Following are the available encoding formats:

1. CDIF (Corpus Document Interchange Format) which was used by BNC (British National Corpus).
2. CES (Corpus Encoding Standards) which is used by EMILLE project (Enabling Minority Language Engineering).
3. XCES (XML Corpus Encoding Standards) which was used by ANC (American National Corpus).

The format that is to be used for the Dzongkha Text corpus is CES (Corpus Encoding Standards) [6] as it is convenient to be used for minority language like Dzongkha.

2.3. Corpus Cleaning and Processing

The clean up stage involves checking whether the corpus actually works the way it ought to, and fixing it where it doesn't. The purposes of this stage are to test the accuracy of what you learned during the information gathering stage (which often turns out to be inaccurate); to get a basic grasp of the approach you'll need during conversion, and to manually fix the problems that would be difficult to handle during conversion.

i) General problems in raw corpus and cleaning issues

- Duplication of words and also sentences. (This means repetition of words).
- Due to lack of web pages of minority language like Dzongkha, most texts are written in electronics text form manually in which human error can occur.
- Existing of foreign (English) language words in between the texts.
- Lack of spell checker cause spelling mistakes
- Texts as a whole is not balanced (Mostly in case of genre).
- Statistical analysis such as word level frequency analysis has to be done manually which is time consuming.

ii) Available tools and techniques for cleaning and processing of the text corpus

-Manual correction of spellings due to lack of spell checker.
-Deletion of unwanted spaces in between texts using Perl command.

2.3.1. Cleaning and Data Extraction Process

2.3.1.1. Extraction Process

A tool for the extraction of texts from the websites such as web crawler was available but we have done the process of copying manually from the websites (due to lack of website in Dzongkha) by copy-paste process. It was easier for us since only a few numbers of websites are available in Dzongkha.

The selection of text from the website and electronic media (mostly from Media) was made by studying the richness in linguistic structure of the text and considering the variety of domains required.
We also got copyright for all the text from respective sources so that we can redistribute it later.

2.3.1.2. Cleaning process

Our text cleaning process is divided into four major steps:-

- Most of the text was typed manually into electronic form, so correction of manual mistakes such as spelling correction, unwanted texts, & repetition of words were removed.
- Paragraph boundaries are marked as required following the encoding standard (CES corpus encoding standard). Such boundaries are difficult to identify on web text as it does not always consists of grammatical sentences.
- Finally tokenization of words and marking of sentence boundaries was done using Dzongkha tokenizer (developed using python).

Also Perl command was used to delete unwanted white space and tabulators.

*Tokenization of words is required for the POS tagging of each word and for frequency analysis of words.

2.3.2. Problems and Drawbacks

Word Segmentation

The major problem faced at present is the unavailability of a word segmentation tool for our language Dzongkha. For POS tagging the text needs to be segmented to the word level. Solution: Till now for creating the training corpus, 20,000 words were manually segmented. After developing automatic tagger for Dzongkha using Tree tagger tool, the input text to the automatic tagger is required in one-word per line format. For which tokenizer tool was used, this segments the text to syllable level. A word segmentation tool would be more convenient than a tokenizer.

2.3.4. Algorithms

i) Tokenizer tool: It functions the same as the tool explained in the Tokenization process as is explained in the latter topics (see 3.2.2).

ii) Tree tagger Tool

We have four following steps for performing the tagging:-

- Manual POS tagging of 20,000 words for training corpus.
- Creation of parameter file, dzongkha.par from training corpus, lexicon file, and open class file.
- Automatic tagging was performed.
- Calculation of accuracy through cross validation process.

Ongoing work involves doing automatic tagging and manual correction to the output file from it.

Drawbacks: The accuracy of Tree tagger tool was around 85% which is not very good. It involves a lot of manual correction which is very time consuming.

Solution: Automatic tagging increases the no. of tagged data which eventually improves the accuracy of the tagger. So aim is to increase the accuracy from 85% to 95%.

Also the Tree Tagger tool requires a word segmentation tool since input to the tagger has to be

done in one-word per line format.

2.4. Additional Considerations

As of now, we have not experienced any conflict resulting from licensing issues since the texts are mostly collected from freely available sources and some are acquired through written permissions from the concerned agencies. For instance Dzongkha Development Commission issued a formal written permission for using their dictionaries and other relevant linguistics resources [7, 8, 9, 10]. As long as the use is confined to academic purposes, there will not be any issues even on distribution of corpora database.

2.4.1.

Dzongkha has a unique problem of word segmentation and has different word counting. The words are separated by the syllable marker called “Tseg” represented as
“ ་ ”. The examples are as follows:

a) རྒྱལ་ཁབ་

“Country”

Here, the two separate syllables combine to form one word. In this instance only one word is counted (as per the meaning), while there's also the instance of considering just one syllable as one word as in the example below.

b) ནང་

“In”

Therefore, the counting system is not consistent in Dzongkha. In order to disambiguate the problem, we need a system which can recognize an accurate word.

In future, we intend to research on designing the spoken component, which will include important spoken conversation and Parallel corpora.

3. Word Segmentation and Tokenization

3.1 Word Segmentation

3.1.1. Description of problem

Segmentation of Dzongkha word is a problem for the related natural language processing tasks such as Spell Checker. Therefore, it is important to segment the word to process the Dzongkha text. Similar to other Asian Languages, Dzongkha words have no explicit word delimiter such as spacing in English to indicate the word boundaries. The most challenging feature of a Dzongkha sentence is the lack of separation between the words. A Dzongkha word is made up of one or more syllables. The delimiter available in Dzongkha script called Tsheg(་) is used to separate the syllables. A syllable can be of single-character syllable, double-character syllables, triple-character syllables and quadruple-character syllables. And a word can be one syllable, two syllables and multi-syllables. [see 2.4.1]

3.1.2. Current Work on Word Segmentation and Methods

The present work on word segmentation is still in progress, the work was started very recently.

i) Methods

Lexicon based Maximal Matching is the method we use in our current work on Dzongkha word segmentation. Maximal Matching techniques generate all possible segmentation from a given input sentence based on provided lexicon. Reading the input string, the maximal matching method generates all possible segmentations. Bigram technique is then applied to select the best segmentation.

For example,

If the given input sentence is: འདི་རྫོང་ཁ་གི་ཞིབ་འཚོལ་ཡིག་ཆཨིན། (meaning, “This is research document of Dzongkha”), then applying the method described produces following possible segmentations, where it is sorted on its weight (highest weight first) given during the lexicon lookup. The number of possible segmentations depends upon the value of n (tree pruning threshold).

1. འདི།རྫོང་ཁ།ཀྱི།ཞིབ་འཚོལ།ཡིག་ཆ།ཨིན།

this | Dzongkha | of | research | written document | is

2. འདི།རྫོང་ཁ།ཀྱི།ཞིབ་འཚོལ།ཡིག།ཆ།ཨིན།

this | Dzongkha | of | research | (generalized term for label/tag/list/address, etc) | 6th dzongkha script | is

3. འདི།རྫོང་ཁ།ཀྱི།ཞིབ།འཚོལ།ཡིག་ཆ།ཨིན།

this | Dzongkha | of | arrange together | search/expose | written document | is

4. འདི།རྫོང་།ཁ།ཀྱི།ཞིབ་འཚོལ།ཡིག་ཆ།ཨིན།

this | fortress | (mouth/surface/language/2nd dzongkha script) | of | research | written document | is

5. འདི།རྫོང་ཁ།ཀྱི།ཞིབ་འཚོལ།ཡིག།ཆ་ཨིན།

this | Dzongkha | of | research | (generalized term for label/tag/list/address, etc) | (unknown word)

6. འདི།རྫོང་ཁ།ཀྱི།ཞིབ།འཚོལ།ཡིག།ཆ།ཨིན།

this | Dzongkha | of | arrange together | search/expose | (generalized term for label/tag/list/address, etc) | 6th dzongkha script | is

7. འདི།རྫོང་།ཁ།ཀྱི།ཞིབ་འཚོལ།ཡིག།ཆ།ཨིན།

this | fortress | (mouth/surface/language/2nd dzongkha script) | of | research | (generalized term for label/tag/list/address, etc) | 6th dzongkha script | is

From the above generated possible segmentations, we select the best and appropriate segmentation, by comparing with the provided word delimited text corpus. This method is called Bigram Techniques [13, 14].

3.1.3. Algorithmic Details

(a) Maximal Matching

Step 1: Read the input of string text. If an input line contains more than one sentence, a sentence
separator is applied to break the line into individual sentences.

Step 2: Split input string of text by Tsheg(་) into syllables.

Step 3: Taking the next syllables, generate all possible strings

Step 4: If the number of strings is greater than (n) where n is the tree pruning threshold.

a) Check the word count of each string from Lexicon.
b) Sort the string on number of words
c) Delete (number of strings – n) low count strings

Step 5: Repeat from Step 2 until all syllable are processed.

(b) Bi-gram Techniques [13, 14]

The Bi-gram Techniques is used to decide the most appropriate segmentations from the list of possible string generated by Maximal Matching method. For this technique a word delimited text corpus is required.
With Bi-gram Technique we can calculate the probability of next word given the previous word. That is,

$P(w_i / w_{i-1})$

where

$$P(w_i / w_{i-1}) = Count(w_{i-1} w_i) / Count(w_{i-1})$$

where

$Count(w_{i-1} w_i)$ total occurrence of word sequence $w_{i-1}w_i$ in the corpus, and
$Count(w_{i-1})$ total occurrence of word w_{i-1} in the corpus.

3.1.4. Drawbacks

The Dzongkha lexicon size is approximately over 20,000 head word entries. The above method mostly produces proper segmentations when all the words are covered by the

lexicon. But, if there are definition of new words, that are not fully covered by lexicon, this techniques generate wrong word boundaries.
Similarly, we have over 0.4 million collected corpus, but only few have been trained and delimited for the use in Bi-gram Techniques. If we have more coverage of trained corpus, higher accuracy of word segmentation is guaranteed.

3.1.5. Conclusion and Future Works

The work is in the initial stages still in progress where there are lot of improvements to be done like the improvement of the overall performance optimization of word segmentation. Future work will include research on ways to handle ambiguous word boundaries and solutions for detection of unknown words.

3.2. Tokenization

3.2.1. Description of problem

In Dzongkha the basic unit for writing is a syllable. There is no inter word space and each syllable is delimited by a syllable marker called a “tsheg” represented as “ ་ “. Sometimes one syllable makes up a word and other times two or more syllables are taken together as a word. There is no rule as to how many syllables are required for the formation of a word. It depends from a particular word to another.

The fact that there is a syllable marker makes segmentation of syllables easier in a sentence. Hence, words can be split using the syllable marker. We also have a sentence marker/paragraph marker represented by “ ། ” called 'shed'. It basically marks the end of an expression.

In Dzongkha Text to Speech (TTS) system, Grapheme to Phoneme (G2P) conversion is implemented using a look up dictionary. The dictionary just consists of Dzongkha syllables along with its pronunciation [15, 16].

Example of a Dzongkha sentence with its pronunciation and meaning:

ཁྱོད་ག་སྟེ་ལུ་འབདཔ་སྨོ?

ch-oe-x-0|g-a-x-0|t-e-x-0|l-a-x-0|b-eo-x-0|m-o-x-1|*
you where work do?
The example sentence consists of seven syllabbles with each one of them seperated by a syllable marker " ˙ ".

Abbreviation and Numeral expansion

Dzongkha text normalization process involves normalization of symbols, dates and numbers.

There are many symbols in text corpus. They are mainly used for text decoration and do not have pronunciation. These symbols (ཀ, ཾ, -, :-,.) need to be removed.

Dzongkha does not have a different way of speaking dates like in English, for example 24 -> TWENTY FOUR is a number whereas 24th -> TWENTY FOURTH is a date, but there are tokens (syllables), in front and in between that indicates it is a date.

If the date is in short form like in English, 1/12/09, which is equivalent to ༡/༡༢/༠༩ in Dzongkha, this has to be expanded to its standard form, that is "སྤྱི་ལོ་ ༢༠༠༩ སྤྱི་ཟླ་ ༡༢ ཚེས་སྤྱི་ཚེས་ ༡ ".

Similarly, digits have to be converted into its letter form before any processing can be done on the text.

For example: below is a standard date form which is converted to its equivalent letter form in the normalization process before it is passed on for G2P conversion.

སྤྱི་ལོ་ ༢༠༠༩ སྤྱི་ཟླ་ ༡༢ ཚེས་སྤྱི་ཚེས་ ༡

སྤྱི་ལོ་ གཉིས་སྟོང་ལེབ་དགུ་ སྤྱི་ཟླ་ བཅུ་གཉིས་ ཚེས་ སྤྱི་ཚེས་ གཅིག་

c-i-x-0|l-o-x-0|ny-i-x-1|t-o-ng-0|l-e-b-0|g-u-x-0|c-i-x-0|d-a-x-0|c-u-x-0|ny-i-x-1|p-ai-x-0|c-i-x-0|tsh-e-x-0|c-i-x-0|*

3.2.2. Algorithms

a) Literature review on work done for the language

This is the first research work, as far as we know and to our knowledge, ever conducted and documented for the language.

b) Current Work

The Tokenizer is based on syllables. The step wise algorithm is shown for Dzongkha Tokenizer.

Step 1. Read from a text file
Step 2. Store it in a variable
Step 3. Break it down into sentence (by the sentence marker “།”)
Step 4. Store all sentence in a list
Step 5. Process the list sentence by sentence (list element)
Step 6. Split the sentence by syllables (by the syllable marker “་”)
Step7. Store the syllables of each sentence in a separate list
Step 8. Repeat Step 5 to Step 7 till end of sentence list
Step 9. Pass the syllables sentence wise to Normalization module to check for Date and number tokens.
Step 10. Normalize Date and number tokens.
Step 11. Pass Normalized tokens to Grapheme to Phoneme module.

Sample Input:

༡༦ སེམས་རྟོགས་ཀ་རིག་གཞུང་མཐོ་རིམ་སློབ་གྲྭ་ལས་ དུས་ཡུན་ཆུང་ངོ་༢ ཀྱི་རིངས་ལྟ་ གཅིག་བསིལ་ཉམས་དང་ལྡན་པའི་སྟོང་ཕུག་མིང་གི་ཚངས་རང་ལས་ ཀ་ཡར་འཛོགས་ཏེ་ འགྱོམ་དུ་ མཐོ་ཚད་མི་ཉར་ ༢༥༦ དང་ལྡན་པའི་ས་གནས་ ལྷག་ལ་ཀ་ལྟ་ལྟོད་པ་ཞིན་ས་དང་ ཁུར་ཆ་འདི་ཆུ་ཡང་ རྐ་གུ་བཀལ་ཏེ་འབག་འགྱོ་ནི་ཡོད་པ་ཞིན།

ལྷག་ལ་ཀ་ལས་ ཀ་གྲེན་འཛོགས་ཏེ་ སྟོང་ཕུ་དང་ཨིང་སོགས་ཀྱི་སྒྲུག་ལས་འགྱོ་སྟེ་ ཆུཆོད་གཉིས་དང་ཕྱེད་ དེཅིག་གི་ས་ཁར་ མཐོ་ཚད་མི་ཉི་ ༥୦୦୦ འབད་མི་བྱི་ལི་ཇི་ལ་ལུ་ལྟོད་པ་ཞིན།

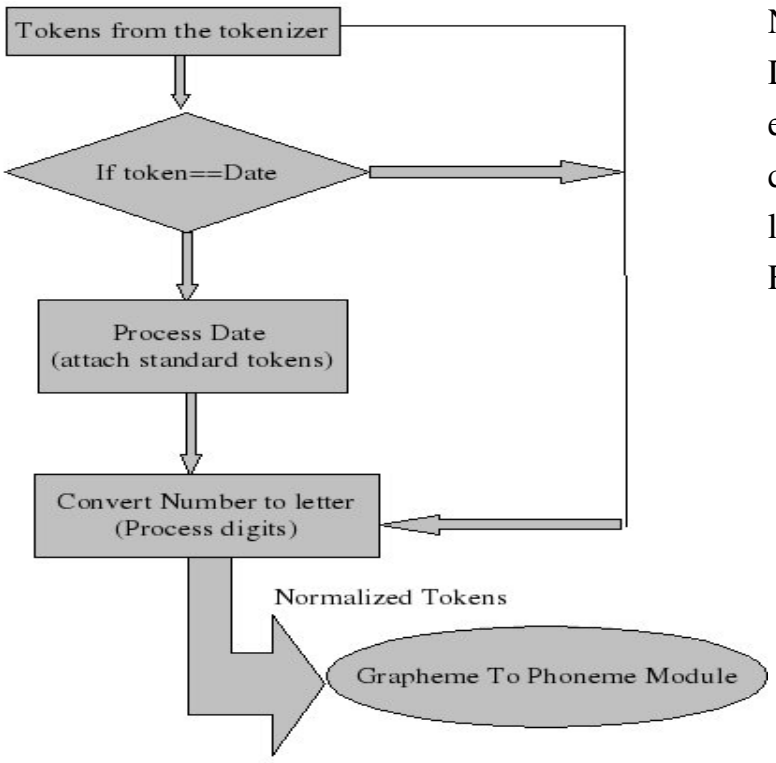
The sentences are first broken down. And then the syllables or tokens from each of the sentences are separated. In the above sentence “སེམས” is the first token as it is separated from the second token “རྟོགས” by a syllable marker “་” . Also the dates and number tokens are normalized to get their representative tokens (equivalent text representations). The resulting list of tokens are then passed onto a Grapheme to Phoneme convertor.

Output:

c-i-x-0|s-e-m-0|t-o-x-0|kh-a-x-0|r-i-g-0|zh-u-ng-0|th-o-x-0|r-i-m-0|l-o-b-1|dr-a-x-0|l-e-x-0|d-ue-x-0|y-ue-n-0|ch-u-x-0|tsh-oe-x-0|ny-i-x-1|k-i-x-0|r-i-ng-0|l-u-x-0|g-a-x-0|n-i-x-0|b-a-x-0|s-i-x-0|ny-a-m-0|d-a-ng-0|ng-e-n-1|p-ai-x-0|t-o-ng-0|ph-u-x-0|sh-i-ng-0|g-i-x-0|tsh-a-ng-0|n-a-ng-0|l-e-x-0|kh-a-x-0|y-a-r-0|dz-e-g-0|t-e-x-0|j-ou-x-0|d-a-x-0|th-o-x-0|tsh-e-x-0|m-i-x-0|tr-a-r-0|ny-i-x-1|t-o-ng-0|g-u-x-0|j-a-x-0|ny-e-r-0|ng-a-x-1|d-a-ng-0|ng-e-n-1|p-ai-x-0|s-a-x-0|n-e-x-1|t-a-x-0|l-a-x-0|kh-a-x-0|l-u-x-0|lhh-oe-p-0|@-i-m-0|d-a-ng-0|kh-u-x-0|ch-a-x-0|d-i-x-0|tsh-u-x-0|y-a-ng-0|t-a-x-0|g-u-x-0|k-e-l-0|t-e-x-0|b-a-x-0|j-o-x-0|n-i-x-0|y-oe-p-0|@-i-n-0|*

t-a-x-0|l-a-x-0|kh-a-x-0|l-e-x-0|kh-a-x-0|j-e-n-0|dz-e-g-0|t-e-x-0|t-o-ng-0|ph-u-x-0|d-a-ng-0|b-a-x-0|sh-i-ng-0|s-o-g-0|k-i-x-0|b-u-x-0|l-e-x-0|j-o-x-0|t-e-x-0|ch-u-x-0|tsh-oe-x-0|ny-i-x-1|d-a-ng-0|ch-e-x-0|d-e-x-0|c-i-x-0|g-i-x-0|s-a-x-0|kh-a-r-0|th-o-x-0|tsh-e-x-0|m-i-x-0|tr-a-r-0|zh-i-x-0|t-o-ng-0|b-e-x-0|m-i-x-0|j-i-x-0|l-i-x-0|dz-i-x-0|l-a-x-0|l-u-x-0|lhh-oe-p-0|@-i-n-0|*

Eventually all the tokens are transcribed automatically with their pronunciation (Grapheme to Phoneme Convertor). The * indicate end of sentence. The '|' marks the end of a syllable. The 0 and 1 are normal and high tones respectively. The x stands for the missing final consonant. Since the synthesis is based on the format “initial consonant-vowel-final consonant-tone”, where there is no final consonant in a syllable, x takes the place of the final consonant so that the format is preserved. ('x' does not have pronunciation)



Normalization of Dzongkha consists of expansion of dates and converting digits to letter form as shown in Figure 3.1.

Figure 3.1. Expanding Dates

A pattern matching is done to find date tokens by the date processor. If the token matches and returns true for date, the year token is converted to its standard form of four digits instead of two. After that, the month and the day tokens are appended in the front of the year so that it is in its expanded form. Once it is in its standard form the digits are converted to its equivalent letter for processing.

Sample code for matching ๑/๑๒/๐๕

x=re.compile("[๐๑๒๓๔๕๖๗๘๙]*/[๐๑๒๓๔๕๖๗๘๙]*/[๐๑๒๓๔๕๖๗๘๙]*", re.L)
re.search(x, token)

Input to Date processor: ๑/๑๒/๐๕
Out put: སྤྱི་ལོ་ ๒๐๐๕ སྤྱི་ཟླ ๑๒ ཐའི་སྤྱི་ཚེས ๑
Input To the Number to letter Processor: སྤྱི་ལོ་ ๒๐๐๕ སྤྱི་ཟླ ๑๒ ཐའི་སྤྱི་ཚེས ๑
Out put: སྤྱི་ལོ་ གཉིས་སྟོང་ལེབ་དགུ་ སྤྱི་ཟླ བཅུ་གཉིས་ ཐའི་ སྤྱི་ཚེས་ གཅིག་
Input to Grapheme To Phoneme Module: སྤྱི་ལོ་ གཉིས་སྟོང་ལེབ་དགུ་ སྤྱི་ཟླ བཅུ་གཉིས་ ཐའི་ སྤྱི་ཚེས་ གཅིག་
Out put: c-i-x-0|l-o-x-0|ny-i-x-1|t-o-ng-0|l-e-b-0|g-u-x-0|c-i-x-0|d-a-x-0|c-u-x-0|ny-i-x-1|p-ai-x-0|c-i-x-0|tsh-e-x-0|c-i-x-0|*
Number to letters

A number token is read and kept aside. It is then broken down by its place value, like a four digit number has four places. For each place processing is done in a different function. So for units place, a different processing is done from a tens place and so on. The processing actually looks up the digit in a already stored python dictionary to get its equivalent letter from. Below are some of the python dictionaries created for Dzongkha.

dzo_eng_digits = {u"๐":0,u"๑":1,u"๒":2, u"๓":3, u"๔":4, u"๕":5, u"๖":6, u"๗":7, u"๘":8, u"๙":9}

digits = {u"๐":u"",u"๑":u"གཅིག", u"๒":u"གཉིས", u"๓":u"གསུམ", u"๔":u"བཞི", u"๕":u"ལྔ", u"๖":u"དྲུག", u"๗":u"བདུན", u"๘":u"བརྒྱད", u"๙":u"དགུ"}

tens_helper = {u"๑":u"བཅུ", u"๒":u"ཉེར", u"๓":u"སོར", u"๔":u"ཞེ", u"๕":u"ལྔ", u"๖":u"ཉོ", u"๗":u"དོན", u"๘":u"བཀྲ", u"๙":u"གོ"}

place_word = [u"", u"", u"བརྒྱ", u"སྟོང", u"ཁྲི"]

tens_place_word = {u"๑":u"བཅུ", u"๒":u"", u"๓":u"ཅུ", u"๔":u"བཅུ", u"๕":u"བཅུ", u"๖":u"ཅུ", u"๗":u"ཅུ", u"๘":u"ཅུ", u"๙":u"བཅུ"}

The dzo_eng_digits is just for comparision of a Dzongkha digit to an English one. From digit dictionary we get each digits letter form. The tens helper is for those number in tens place. In Dzongkha if the number is ๑๒, we say བཅུ་གཉིས, so now the actual letter form is a combination of tens_helper and digits as shown above in the dictionary. Like wise for all digits, Its a combination of these above dictionary items. The combination depends on the place the digit occupy in the number, unit place, tens place and so on.

The process of normalizing numbers into letters is shown in Figure 3.2.

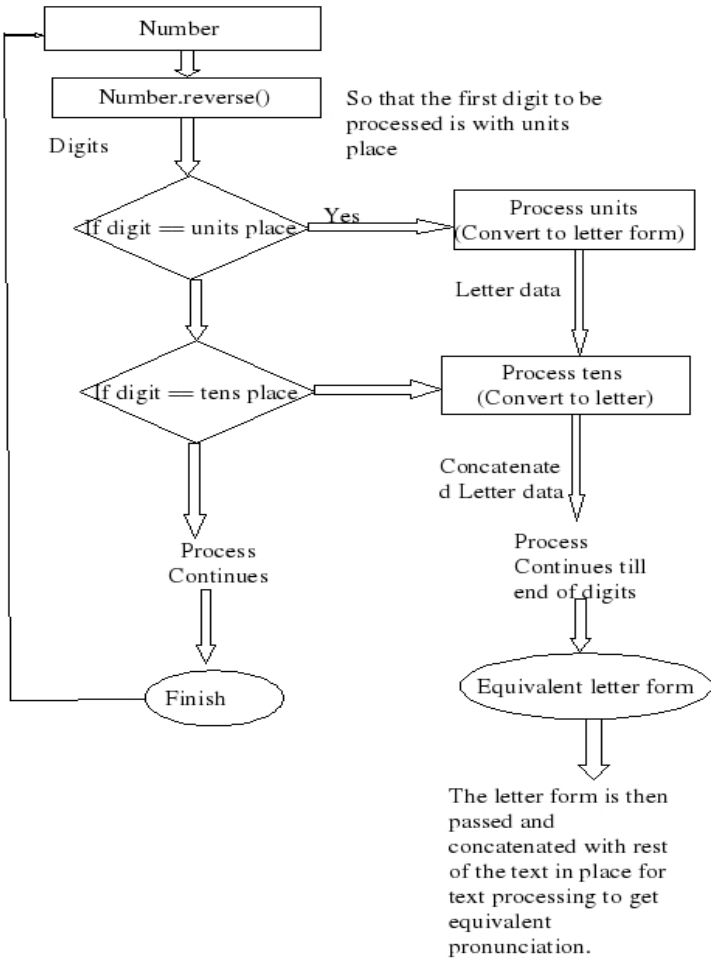


Figure 3.2. Number to Letter Conversion

c) Applications

On the basis of the above defined algorithm, a Dzongkha Text To Speech Synthesis (TTS) [5] has been built. These above algorithms mainly consist of the text analysis module in the TTS system. The text analysis part consists of the syllabification algorithm and Grapheme to phoneme module right now, the normalization part has yet to be integrated in the TTS system. The speech synthesis module is employed using the Hidden Markov Model based speech synthesis system. This TTS is at the moment available for Linux and windows operating system.

As a whole the TTS gives a Mean opinion score of 3.19 as opposed to human speech which was scored at 3.93.

The accuracy of the text analysis module (syllabification, normalization and grapheme to phoneme conversion) is given in Table 1:

Table 3.1. Results of Text Analysis Module

No. of sentence	No. of syllables	No. of Non Standard Words	No. of tokens outputted	No. of Non Standard words correctly normalized
5	216	8	226	8
50	352	1	347	1
100	739	1	731	1
500	3469	3	3444	3

The difference in number of tokens input and the number of tokens output is because the non-standard words are first expanded before tokenization and this causes in the number of tokens outputted to be slightly more in the first case. There is an aggregate of 98% accuracy when a normal Dzongkha text is inputted.
As long as the syllables are marked correctly without any mistakes (spelling) then the tokenizer will be accurate. It's main aim is to separate the syllables using the syllable marker.

3.2.2. Future Work

In future, normalization has to be integrated in the text analysis module, an automatic token labeler may be investigated so that the normalization will become more accurate and better. Word and phrase detection algorithms also may be explored to improve the accuracy and performance.

4. Lexicon

Dzongkha Lexicon has entries made in an excel sheet. It is written in capital letter of Dzongkha (called Tshuyig or Bhutanese formal longhand 'jotshum'). The size of the lexicon is approximately 22,747 word entries.

4.1. Linguistic analysis and design

Sources for lexicon were derived from the available dictionaries published by Dzongkha Development Commission such as Dzongkha–English Dictionary [7], English-Dzongkha Dictionary [8] and a new Dzongkha Grammar [9] and also from the Advanced Dzongkha Dictionary published by KMT [11]. While studying these dictionaries, it was found that most of these dictionaries had no other linguistic details apart from the basic part-of-speech.

Therefore, the Dzongkha Lexicon which is currently being designed and developed will include all the lexical entries from these dictionaries with linguistic details such as part-of-speech, pronunciation symbols based on IPA, pronunciation guides, definitions, examples, cross references, etc.

In addition, we also intend to include the new Dzongkha Computer Terms [12]. The Dzongkha Lexicon will be implemented using XML and scripting language or using the existing application like WordNet.

Some examples from language for various POS categories are as follows:

1. Noun (མིང)
   Dzongkha= ཀ་ཀུ་རྩ། (མིང) ཚོར་བསྡེཅག་གི་མིང། དཔེ - ཀ་ཀུ་རྩ་གི་ཚོར་མ་འདི་ཞིམ།
   English= Pumpkin (Part of speech: Common Noun Tag: NN)

2. Verb (ཐ་ཚིག)
   Dzongkha = འབྱི/བྱས། རྡོ་རྗེ་དེ་ཚབྱས་དེས/ རྡོ་རྗེ་གིས་དེ་ཚའབྱི་ནི་ཞིན།
   English = 'Dorji has done the work' (Part of speech: Verb Tag: VV)

3. Adjective (མིང་གི་ཁྱད་ཚིག)
   Dzongkha= རིངམོ། (མིངཁྱད) ཤིང་འདི་ རིངམོ་ འདུག
   English = 'The tree is tall' (Part of speech: Characteristic adjective Tag: JJCt)

4. Adverb (བྱ་བའི་ཚིག་ཆུང་ངོ་)

Dzongkha= མཚོག་གས། (བྱ་ཆུང་) རྟ་ལས་ སྐྱམ་འགོངས་ མཚོག་གས།

English= 'The vehicle is faster than the horse'(Part of speech: Comparative adverb Tag: RBR,)

5. Conjunction (འབྲེལ་ཚིག)

Dzongkha = རྩད། རོ་རྒྱན་བཀོག་རྩད་ ལས་བསགས་པ་ཞིན།

English= It is a sin, even to throw a stone in the water (POS: Subordinate Conjunction, Tag: SC, Category: Conjunction)

6. Preposition (ཚོག་ཐད)

Dzongkha-འོག ཌི་ལི་ ཤིང་གི་འོག་ལུ་འདུག།

English- A cat is under the tree (Tag: PP)

4.2. Current Work

Currently, lexical entries are formatted as a simple excel sheet data. Later, it will be transferred into XML database, SQL database or they can be imported by WordNet application, based on the choice of design platform and technologies. As of now, there are approximately 22,747 word entries in the lexicon.

Word classification, assigning POS tags without ambiguity, assigning accurate phonetic symbols, categorization of words into synonyms, antonyms, assigning stress and tone markings are some of the challenges encountered in designing the lexicon.

4.3. Word Net

Preliminary study on WordNet has just started. We are currently studying the basic features of WordNet and its functional capabilities. Existing WordNet specially the Princeton WordNet is being researched to understand the building of WordNet database, aspects of word sensing and incorporation of linguistic details into WordNet.

5. Part of Speech

5.1. Tagset

i. Previous linguistic and computational linguistic works by others

As far as we know, we are not aware of any computational linguistic works done by any individual or an organization. But, there are few linguistic researches done by the Dzongkha Development Commission especially on grammar [10] and dictionaries [7, 8, 9]. The monolingual and bilingual dictionaries [7, 8, 9, 10] are some of the important linguistic work published by them. George Van Driem, in collaboration with Karma Tshering, Gaselo, has also done research study pertaining to the Languages of Greater Himalayan Region [1], in which detail linguistic characteristics of Dzongkha has been studied. These includes important Linguistic characteristics like the origin of the language, character set (consonants, vowels, etc.), tones among others.

ii. Summary of Current tagset

Currently, there are about 42 POS tagset in Dzongkha as summarized in Table 2.

Table 5.1. Summary of Current tagset

Category	Remarks	POS Tag ID No.	POS Name	POS Tag
Noun	• Most often in Dzongkha, plural can be known by script (ཚུ་ "tshu") and of course (ཆ་ཚང་ "chachap" which means 'they'). • We can of course consider singular noun as common noun but it	1	Common noun	NN
		2	Particular/Personal noun	NNP
		3	Honorific noun	NNH
		4	Quantifier noun	NNQ
		Sub-tags		
		i)	Singular noun	NNs

	cannot be a common noun always. Therefore, they are separately tagged.	ii)	Plural noun	NNS
Verb	• Auxiliary and agentive are different in Dzongkha, where auxiliary acts as a helping verb and agentive as cause or agent of an action. Sometimes agentive can act as an instrument of an action.	5	Auxiliary	VBAUX
		6	Modal	VBMD
		7	Agentive verb	VBAt
		8	Imperative verb	VBI
		9	Non-agentive verb	VBNa
		10	Aspirational verb	VBAs
Adjective		11	periodic adjective	JJP
		12	Characteristic adjective	JJCt
		13	Comparative adjective	JJR
		14	Superlative adjective	JJS
Adverb		15	Behavioral adverb	RBB
		16	Comparative adverb	RBR
		17	Superlative adverb	RBS
Pronoun	iii. Just like English, Dzongkha also has first; second and third person: I=First; you=second; He+She+They=third person.]	18	Personal pronoun	PRP
		19	Differential pronoun	PRD
		20	Reflexive pronoun	PRRF
		21	Locative pronoun	PRL
		Sub-tag		

	• In Dzonkha, locative pronoun and postposition are very similar, although syntactically different. • Locative pronoun is just a pronoun for any places(like, there; here; up etc.) • Postposition occurs after noun. Its role is same with English preposition(like, under; on; beside etc.) Similarly, ablative case, coordinate conjunction(ལས་ only) and range(adposition) look very similar: a. Ablative case(CA) indicates a source or origin of a person and an object. b. Coordinate conjunction(CC) serves to conjoin words or phrases or clauses or sentences. c. Range(PRa) appears like preposition as it forces a nominal argument to follow. It is clearly positioned between two nominal numbers. ci.	i)	Range	PRa

Case	• Genitive case is almost same like postposition but it marks only link in a phrase.	22	Genitive case	CG
		23	Vocative case	CV
		24	Dative case	CDt
		25	Ablative case	CA
Conjunction		26	Subordinate conjunction	SC
		27	Coordinate conjunction	CC
Postposition		28	Postposition	PP
Determiner	Possessive article is always attached with a genitive case in a phrase, attributing possession to someone or something.	29	Definite article	DT
		30	Indefinite article	DTI
		31	Demonstrative article	DTdm
		32	Possessive article	DT$
Tense	Note: In Dzongkha, we have two tenses in future, two tenses in present and two in past. Future: 'Ni' and 'Wong'=(...will/shall etc..) Present: 'D'o' and 'D'ä'=(...ing etc..) Past: 'Yi' and 'Ci'=(went,gone,did,finished etc..)	33	Tense marker	TM
Interrogation	There are four interrogative marks: 'ga','na','ya' and "mo'=?(for English) In Dzongkha, the interrogative mark is optional, we either use those consonants or question mark(?).	34	Interrogative mark	IrM
Affirmation		35	Affirmative mark	AM
Head mark		36	Head mark	HM
Punctu		37	Punctuation	PUN

ation				
Foreign word		38	Foreign word	FW
Symbol		39	Symbol	SYM
Unknown word		40	Unknown word	UKW
Cardinal Number		41	Cardinal Number	CD
Ordinal Number		42	Ordinal Number	OD
Nominal Number		42	Nominal Number	ND
Interjection		44	Interjection	UH
Negator		45	Negator	NEG

iv. Detailed explanation of tags with examples

1. Examples
    - *Common Noun (NN)*

a. ཤིང་འདི་ སྒོམ་འདུག།
b. tree this big be
c. “This tree is big”

- *Personal Noun (NNP)*

a. འཇིགས་མེད་སེང་ཆེ་དབང་ཕྱུག་འདི་ འབྲུག་གི་རྒྱལ་པོ་ཨིན།
b. Jigme-Singye-Wangchuk the Bhutan's King be
c. “Jigme Singye Wangchuk is the King of Bhutan”

- *Quantifier Noun (NNQ)*

a. ངཱིས་ བུམོ་ལྔ་དང་ཕྱད་ཏེ།
b. I-aux girl five with meet-tm(tense marker)
c. “I met with five girls”

Sub-tag: Singular noun(NNs)

a. ཀི་དེབ
b. book
c. “A book”

Plural noun(NNS)

a. ཀི་དེབ་ཚུ།
b. Books
c. “The books”

- *Honorific Noun (NNH)*

a. ཨི་དབང་རྒྱལ་པོའི་ཡབ།
b. King-CG father
c. “King's father”

- Ergative/Auxiliary verb (VBAUX)

a. རྡོ་རྗེ་གིས་ ལྷ་ འབད་ནུག།
b. dorji-aux work do-tm
c. “Dorji did the work”

- *Modal verb (VBMD)*

a. ཁོད་ཀྱིས་ ལྷ་ རྩ་འགེངས་ འབད་དགོ།
b. you-aux work try do must
c. “You must work hard”

- *Agentive verb (VBAt)*

a. རྡོ་རྗེ་གིས་ བྱི་ཚི་ བསད་རྒྱུག།
b. dorji-aux cat kill-tm
c. “Dorji killed the cat”

- I*mperative verb* (VBI)

a. ཡི་གུ་ བྲིས་ཤིག
b. letter write-imp
c. “Write a letter”

- *Non-agentive verb* (VBNa)

a. རླུང་མ་ འཕྱར་དེས།
b. wind blow-tm
c. “The wind is blowing”

- *Aspirational verb* (VBAs)

a. ང་ དག་པའི་ཞིང་ལུ་ སྐྱེ་བར་ཤོག།
b. I pure-land in born may
c. “May I be born in the Pure Land”

- *Periodic adjective* (JJP)

a. ང་ ནཀ་ཧི་རྒྱལ་ལུ་ འགྲོ་ཡི།
b. I last-year abroad to go-tm
c. “I went abroad last year”

- *Characteristic adjective* (JJCt)

a. ཤིང་འདི་ རིངམོ་ འདུག།
b. tree the long be
c. “The tree is big”

- *Comparative adjective* (JJR)

a. ཤིང་འདི་ རིང་ཤོས་ཅིག་ཡིན།
b. tree the taller-a be
c. “The tree is taller”

- *Superlative adjective* (JJS)

a. ཤིང་འདི་ གནམ་མེད་ས་མེད་ རིངམོ་ཅིག་ཡིན།
b. tree the very very long-a be
c. “This tree is very very tall”

- *Behavioral adverb* (RBB)

a. སོ་ནམ་པ་ཚུ་ རྩ་འགེངས་ཏེ་ ལྷ་འབད་དེས།
b. farmer-pl try-conj work do-tm
c. “Farmers are working hard”

- *Comparative adverb* (RBR)

a. རྟ་ལས་ སྒྱམ་འགོར་ མགྱོགས།
b. horse-comp vehicle fast
c. “A vehicle is faster than the horse”

- *Superlative adverb* (RBS)

a. རྟ་འདི་ གནམ་མེད་ས་མེད་ མགྱོགས་པས།
b. horse this very very fast be

c. “A horse is very very fast”

- *Personal pronoun* (PRP)

a. རིན་ཆེན་ ཁྱོད་ ནང་ཨོག།
b. Rinchen you here come
c. “Rinchen, you come here”

- *Differential pronoun* (PRD)

a. འབྲུག་པའི་མི་སེར་ གཅིག་ཀྱང་ རྫོང་ཁ་ཤེས་དགོ།
b. Bhutanese-gen-people all-every Dzongkha-know-must
c. “Every Bhutanese people must know Dzongkha”

- *Reflexive pronoun* (PRRF)

a. ཁྱོད་རང་ འགྲོ་དགོ།
b. you-self go-must
c. “You must go yourself”

- Locative pronoun (PRL)

a. ནང་ ཕུན་ཚོགས་གླིང་ ལུ་ ཨོག།
b. here Phuntsholing in come
c. “Come here in Phuntsholing”

Sub-tag: *Range(PRa)*

a. གཅིག་བརྒྱ་ལས་ཉིས་བརྒྱ།
b. One hundred-PRa two hundred
c. “Hundred to two hundred”

- *Genitive case*(CG)

a. ཁབ་ཀྱི་རྩེ།
b. needle-gen-tip

c“tip of needle”

- *Vocative case* (CV)

a. དབའ་ནུ་མོག། སྡོམ
b. hey here-come
c. “hey, come here”

- *Dative case* (CDt)

a. རྡོ་རྗེ་ལུ་བྱིན།
b. Dorji-CDt give
c. “Give to Dorji”

- Ablative case(CA)

a. རྒྱ་མཚོ་ལས་རྡོར་བྱ།
b. ocean-Ca jewel
c. “A jewel from the ocean”

- Subordinate conjunction (SC)

a. གཞན་རྒྱ་འདི་གྱུར་ ས་ཡིག་མ་བརྒྱབ་པ་ཅིན་ དོན་དག་མེད།
b. agreement-the-on signature-not-done meaning no
c. “If the agreement is not signed, it has no meaning'”

- *Coordinate conjunction* (CC)

a. ངལ་ པི་སི་གཅིག་དགོ་པས་ ཡང་ཅིན་ སྐྱུ་ག་གཅིག་དགོ་པས།
b. I-for pencil-one-need-be or pen-one-need-be
c. “I need a pencil or a pen”

- *Postposition* (PP)

a. ཀྱི་ལི་ ཤིང་གི་འོག་ལུ་འདུག།
b. cat tree-gen-under-be
c. “cat is under the tree”

- *Definite article* (DT)

a. རྡོ་རྗེ་འདི་ མི་ལེགས་པོ་ཨིན།
b. Dorji-is person-good-be
c. “Dorji is a good person”

- *Indefinite article* (DTI)

a. མི་ལ་ལུ་ཅིག་གིས་ སྨྲས་མས།
b. person-some-indf-aux tell-be
c. “Some people say..”

- *Demonstrative article* (DTDm)

a. ཡ་ཐི་ལྷ་ཁང་འདི་ སྦོམ་འདུག།
b. that-lhakhang-this big-be
c. “That lhakhang is big”

- *Possessive article* (DT$)

a. ངའི་ ཀི་ཏེབ།
b. my-gen book
c. “My book”

- *Tense marker* (TM)

a. ང་ ནངས་པ་ འགྲོ་ཞི།
b. I tomorrow go-tm (future)
c. “I will go tomorrow”

a. ང་འགྲོ་དེ།
b. I go-ing tm (present)
c. “I am going”

a. ང་འགྲོ་ཡི།
b. I go-tm (past)
c. “I went”

- Interrogation (Irm)

a. ཁྱོད་ ཐིམ་ཕུག་ འགྲོ་ནི་ཨིན་ན?
b. you Thimphu go-tm (future)
c. “Are you going to Thimphu?”

- Affirmation (AM)

a. ཀྱི་ཡི་ ཤིང་གི་ཀུ་ འདུག།
b. cat tree-gen-on be
c. “cat is on the tree”

- Negator (NEG)

a. ང་འཇའ་རིལམ་ མེན།
b. I beautiful not[NEG]
c. “I am not beautiful”

- Cardinal number (CD)

a. ༡༠༠ དང་ ༥༠༠
b. 100 and 500[CD]
c. “100 and 500”

- Ordinal number(OD)

a. ༡ས། ༢ས། ༣ས།
b. 1st , 2nd and 3rd [OD]
c. “1st , 2nd and 3rd”

- Nominal number(ND)

a. བརྒྱད་འཛིན་ཨང་ ༡༧༦༤༠༣༧
b. Mobile number 17649037[ND]
c. “ Mobile number 17649037”

- Interjection(UH)

a. སྤུའོ! ཁྱོད་འཇའ་རིས་མོ་མཛེས་མས།
b. Wow![UH] You beautiful look-AM
c. “Wow! You look beautiful.”

- Head mark(HM)

a. ཇི་ སློབ་དཔོན་མཆོག་ལུ།
b. [HM]-dear Sir,
c. “Dear Sir.”

- Punctuation(PUN)

a. ཕྱི་ཁ་འགྲོ། - ། is punctuation here!
b. outside go
c. “Go outside.”

- Foreign word(FW)

a. སི་ཏི་སྟོངས།
b. CD blank

c. 'Blank CD" - here སི་ཌི(CD) is a foreign word

2. Possible Confusing Tags

i. Auxiliary and agentive are different in Dzongkha, where auxiliary acts as a helping verb and agentive as cause or agent of an action. Sometimes agentive can act as an instrument of an action.

Example 1:

a. རྡོ་རྗེ་གིས་ ལཱ་ འབད་རྒྱག།

b. Dorji-aux work do-tm

c. "Dorji did the work"

Example 2:

a. རྡོ་རྗེ་གིས་ བྱི་ལི་ བསད་རྒྱག།

b. Dorji-aux cat kill-tm

c. "Dorji killed the cat"

ii. In Dzongkha, locative pronoun and postposition are very similar, although syntactically different. Locative pronoun is just a pronoun for any places (like, there; here; up etc.). Postposition occurs after noun. Its role is same with English preposition (like, under; on; beside etc.)

Similarly, ablative case, coordinate conjunction(ལས་ only) and range(adposition) look very similar:

a. Ablative case(CA) indicates a source or origin of a person and an object.

b. Coordinate conjunction(CC) serves to conjoin words or phrases or clauses or sentences

c. Range(PRa) appears like preposition as it forces a nominal argument to to follow. It is clearly positioned between two nominal numbers.

(Locative pronoun)

a. ཨ་ཡུན་ཚོགས་གྲོང་ ལུ་ ཨོང་།

b. here phuntsholing in come

c. "Come here in Phuntsholing"

(Post position)

a. བྱི་ལི་ ཤིང་གི་འོག་ལ་ཡོད་དེ།།

b. cat tree-gen-under-be

c. “A cat is under the tree”

(Ablative case)

a. རྒྱ་མཚོ་ལས་རིན་བྱ།

b. ocean-Ca jewel

c. “A jewel from the ocean”

(Coordinate conjunction)

a. ལྷ་འབད་ཞིན་ལས་ འགྲོ་ནི།

b. work do then[CC] go-will

c. “I will go after I finish mu work.”

(Range)

a. གཅིག་བརྒྱ་ལས་ཉིས་བརྒྱ།

b. One hundred-PRa two hundred

c. “Hundred to two hundred”

iii. Genitive case almost looks like a auxiliary verb but it marks only link in the phrase while auxiliary acts as a helping verb.

Example 1:

a. ཁབ་ཀྱི་རྩེ།

b. needle-gen-tip

c. “A tip of needle”

Example 2:

a. རོ་རྗེ་གཤིས་ ལྷ་ འབད་རུག།

b. Dorji-aux work do-tm

c. “Dorji did the work”

5.3. Open issues and Future Work

The current tagset have to be reviewed and formalized together with the DDC. We need to research further on POS tagset, POS tagging algorithm and test it on the corpora database to determine its accuracy and performance.

6. Dzongkha spell checker

Dzongkha words are rather ambiguous in their formation. There are no special character or space (as is the case in written English) separating the words. Special character 'tsheg' are helpful in many ways by marking the syllables but Dzongkha words are not limited to mono-syllabic words. More than one syllable can form a word as indicated in the previous topics. Therefore a proper word-segmentation algorithm has to be designed and implemented before proceeding to develop a Dzongkha spell checker.

Dzongkha stemmer has not been explored as of now. Here also a word segmentation tool for Dzongkha will be required as a pre-requisite to develop a Stemmer for Dzongkha.

7. References

[1] Van Driem, G. and Tshering, K. (Collab), “ Languages of Greater Himalayan Region”, 1998.

[2] Hussain, S., Durrani, N., “Dzongkha”, A Study on Collation of languages from Developing Asia,
PAN Localization Project, Centre for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Pakistan.

[3] http://www.ethnologue.com/Bhutan/

[4] van Driem, George, 1993, “Language Policy in Bhutan,” Paper for conference 'Bhutan: a traditional order and the forces of change', 22-23 March 1993, SOAS, London

[5] Gyatsho, Lungtaen, “Difficulty in Teaching Dzongkha in an English Medium System,” Journal for Bhutan Studies, http://www.bhutanstudies.org.bt/

[6] http://www.cs.vassar.edu/CES/

[7] Dzongkha-English Dictionary, Dzongkha Development Commission, 1990

[8] Dzongkha-Dzongkha Dictionary, Dzongkha Development Commission, 2005.

[9] English – Dzongkha Dictionary, Dzongkha Development Commission, 2006.

[10] A New Dzongkha Grammar, Dzongkha Development Commission, 1999.

[11] Advanced Dzongkha Dictionary, KMT Publishing House, 2004.

[12] Dzongkha Computer Terms, PAN Localization, published by Centre for Research in Urdu Language Processing, NUCES, Pakistan, 2007.

[13] Jurafsky, D., and Martin, J.H., Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2000.

[14] Sok Huor, C., Rithy, T., Hemy R.P., Navy, N., Chanthirith, C., and Tola, C., "Word Bigram Vs Orthographic Syllable Bigram in Khmer Word Segmentation", PAN Localization: Working Papers 2004 - 2007, Page 249, Section 2.2.

[15] Sherpa, U., Pemo, D., Chhoeden, D., Rugchatjaroen, A., Thangthai, A., and Wutiwiwatchai, C. "Pioneering Dzongkha Text-to-Speech Synthesis", International Conference on Speech Database and Assessments, November, 2008.

[16] Sherpa, U., Pemo, D., Chhoeden, D., "Dzongkha Phonetic Set Description", PAN Localization Team, Bhutan