

Digital Pronunciation Dictionary in Bangla for Computer Assisted Language Teaching, E-Learning, and Speech Technology

Niladri Sekhar Dash¹ and Arpita Chakraborty²
 Linguistic Research Unit, Indian Statistical Institute, Kolkata¹
 Intern at LRU, ISI, Kolkata, S/W Developer at R&D Lab, Kolkata²
 Email: ns_dash@yahoo.com¹, c.arpita23@gmail.com²

ABSTRACT

It will be a nice learning experience for the Bangla language learners if an on-line Bangla education system is supported with a Digital Bangla Pronunciation Dictionary (DBPD), which is accessed in classroom and at home, as the case may be, as one of the most useful reference guides for learning standard and acceptable pronunciation of Bangla words. Keeping this idea at background, in this paper, we have made an attempt to report the design architecture of the proposed digital Bangla pronunciation dictionary, which is being developed with a large lexical database of nearly hundred thousand words that are directly obtained from a digital corpus of Bangla written texts as well as from other digital lexical sources available in the language. This is perhaps the first attempt ever made for any of the Indian languages with a mission for serving the Bangla speakers as well as Bangla language learners with better learning resources and devices for the language across the world. The immediate application of the resource is visualized as a tool for e-governance and on-line language teaching where the learners can access this device to address various linguistic purposes including spelling, pronunciation, part-of-speech, meaning, and usage of words.

Keywords: *pronunciation, part-of-speech, transliteration, orthography, IPA, meaning*

1. INTRODUCTION

The Digital Bangla Pronunciation Dictionary (DBPD) that we are engaged in developing is capable of capturing the unique aspects of the pronunciation of the Bangla words with due reference to their parts of speech, meaning, and usage. At the orthography level each and every headword included in this dictionary is represented in two different scripts:

- (a) The Unicode compatible standard Bangla script for the native Bangla speakers and the others who know the language and its script.
- (b) The Indic Roman script marked with diacritics for those people who do not know the Bangla script but know the Indic Roman characters.

On the other hand, the pronunciation of the headwords, following the process adopted in the Oxford English Dictionary, is represented in the following three ways:

- (a) The pronunciation is orthographically presented in the standard Bangla script for them who can decipher the pronunciation of the words with the help of the known Bangla characters.
- (b) The pronunciation is further presented in the International Phonetic Alphabet (IPA) for those people who know IPA but have no knowledge about the utterance of characters used in the Bangla script. The motivation of using the IPA is further triggered for those people who are trying to learn Bangla language at universities and institutes in India and abroad without having direct support from native Bangla teachers.
- (c) The audio output of the pronunciation of words is also provided in this dictionary for the visually challenged people and the primary learners, who can learn the pronunciation of the words just by listening to the audio records attached to the words.

Thus this pronunciation dictionary is meant to address various kinds of the linguistic needs of the language learners and others who can directly use it in their Bangla language learning activities. On the other end, the speech technology people can use this dictionary as a highly useful resource for the development of text-to-speech device for the language. Since the entire lexical database is stored in the server in a very structured and atomic format, the end users can access the dictionary tool through a well-defined web interface described in the following sections of the paper.

Keeping in mind the paucity of space in giving details about the lexicographic and technical issues involved in the compilation and designing of the dictionary of a unique kind, we have mostly concentrated on the technical as well as technological issues involved in developing this dictionary. Lexicographic issues are addressed elsewhere (Dash 2013). In Section 2, we present a short overview of the pronunciation dictionary; in Section 3, we briefly describe the web architecture model adopted for the dictionary; in Section 4, we present the database model that we have used in the dictionary; in Section 5, we discuss in short the performance factor of the dictionary; in Section 6, we argue for using the dictionary both as a stand-alone as well as a web-based device; and in Section 7, we present a short sketch of the beneficiaries who might be interested and profited from the use of this dictionary.

2. OVERVIEW OF THE DICTIONARY

This paper tries to depict an online dictionary tool giving relevant details of a Bangla words searched by the user along with its pronunciation information. The tool is to be hosted on the website of the *Indian Statistical Institute, Kolkata, India* so that everyone can have access at any point of time and from any place in the world. It is also interesting to know that the tool is further designed as an app for mobile and tab in order to reach a larger set of users, who are more comfortable with the smartphones than using a desktop computer to address their lexical queries.

The tool includes a well-formed Word Search Box (WSB) through which a dictionary user can type a Bangla word she wants to search by using the virtual keyboard inbuilt with the device. While typing a word in the interface, the user can see all the relevant words in the list relating to the word she wishes to search for. The entire list of words having similar character strings will be shown in the form of the dropdown interface so that she can select the most

appropriate one. From the dropdown list a user can also search the intended word, and after that, she can select the word by a single click of the mouse. Immediately all the granular details of that corresponding word will appear on the screen of the monitor or the smartphone in a more structured and organized format (Fig. 1).

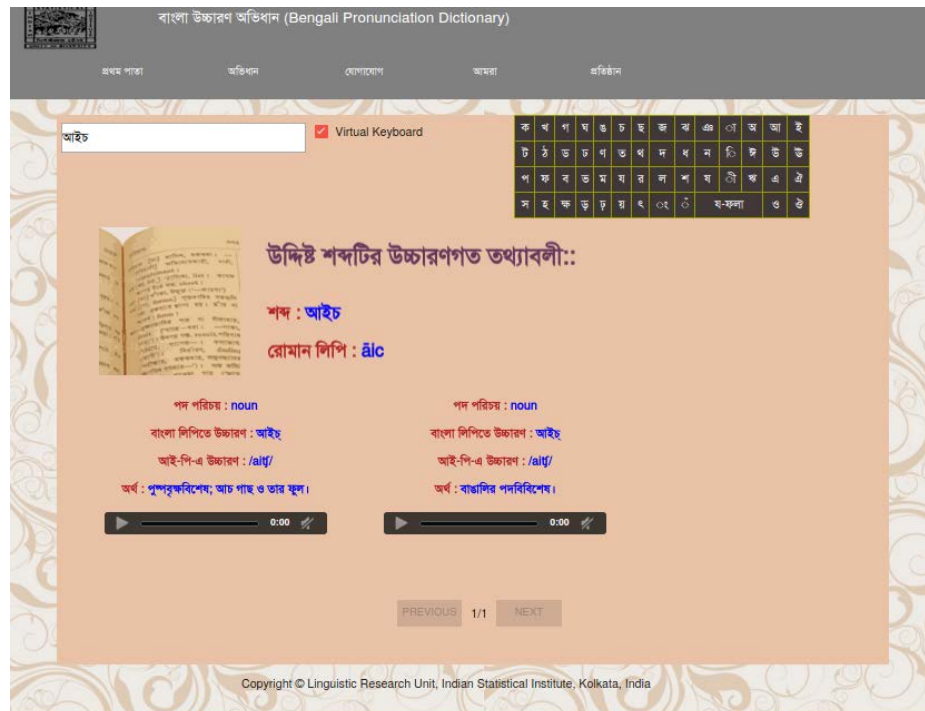


Fig. 1: A screenshot of the GUI of the DBPD

There are many attributes presented in the dictionary relating to the headwords. Among them, the two criteria — word in Bangla orthography and its transliteration in Indic Roman — are the constant factors. That means it has unique and atomic value for that headword searched in the database. The variable attributes, that can contain multiple values for a particular headword, are shown in the grid format with the features of pagination enabled for easy access and enhancing experience of the dictionary users. The attribute list of the headword, which contains more than one value, includes features such as parts-of-speech, Bangla orthography, IPA, pronunciation in audio output, meaning, usage in a sentence, and English translation.

3. WEB ARCHITECTURE MODEL

This on-line Bangla dictionary tool is developed based on a SPA (Single Page Application) architecture that loads a single HTML page and all its related resources one at a time and dynamically updates the partials [i.e., a fragment of a page] as the user interacts with the application. It facilitates the web pages being faster and responsive by avoiding constant page refresh/reload. The Web Architecture model has two parts: (a) Client Side, and (b) Server Side. Client Side is a Web browser. It has two subsections, namely the followings.

- (a) **Master Page:** It contains data template, controls, and sub-layouts that should be shared across multiple partial pages of the site.
- (b) **Partials:** It resides inside the Master Page. Based on the request of a user the inner content will be changed.

In our application, the name of Master Page is INDEX.HTML, which is not visible to the end users but which contains the whole markup, layout template, and design of the application, etc. When the users navigate away from one tab to another the views are updated through changing the state - which means the partials/views content and even its URL will also be changed according to the request of the users.

Example:

User clicks on 1st tab প্রথম পাতা (pratham pata) "First Page"

=> Partial Controller::HomeController, Partial URL::"/Home"

User clicks on 2nd tab অভিধান (abhidhan) "Dictionary"

=> Partial Controller::DictCtrl, Partial URL::"/BengDictionary"

User clicks on 3rd tab যোগাযোগ (jogajog) "Contact"

=> Partial Controller:: ContactController,Partial URL::"/Contact"

User clicks on 4th tab আমরা (amra) "We are..."

=> Partial URL::"/Aboutus"

Note::

Partial URL:: The link to the Partial where content and controls are shown

Partial Controller::The functionalities of the control are handled at its controller.

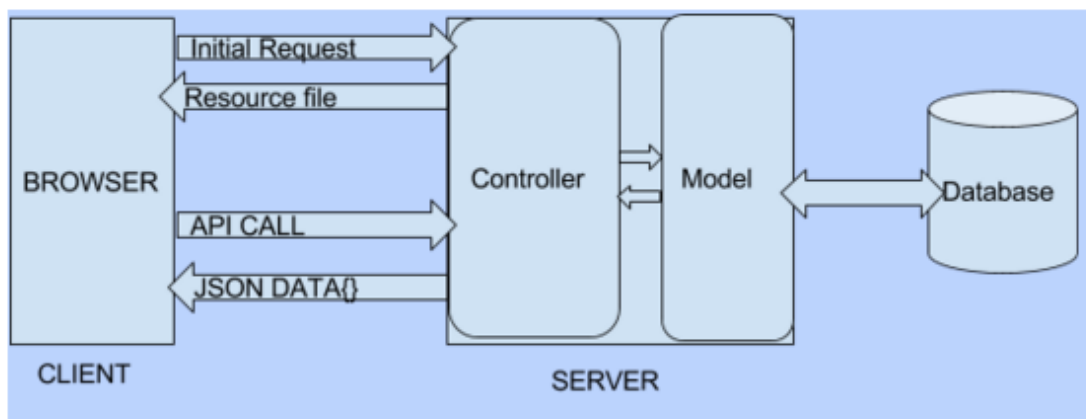


Fig. 2: Client communicates with the server in the form of REST API

As the diagram (Fig.2) shows, the client communicates with the server in the form of REST API which is stateless, asynchronous and uses less bandwidth. It gives output in the form of JSON which is parsed at the browser. Whenever a client makes a first time request to the

server, all the resource files, needed to process, are taken to the browser level to reduce the back and forth communication between the client and the server, which in turn, improves the performance of the web architecture model.

The Server side is actually implemented by forming the API and structuring the class design according to MC (Model-Controller) architecture. The Model contains all the entity classes and performs the data fetching operation from the database. The Entity classes, on the other hand, contain getter and setter method for assigning and fetching the values to the attributes of an entity. There are two entity classes developed for this purpose.

AutoPopulateWord.java: It includes the word attribute information, getter and setter method for executing auto populate functionality.

BengDict.java: It includes all the attribute information associated with the word and getter-setter method to access those attributes.

In our application, there have two API been published.

- (i) **FetchAutoPopulateWordList::** The API is accessed via HTTP GET method passing word as a parameter and returns JSON response with relevant all word names for the requested one. The API will fire after typing one character in the search box and all the relevant words in the list will be displayed in a dropdown box from where the user could select the target word. The drop-down list is gradually getting filtered as the user limits the scope.

URL = relative_url + api/bengwordautopopulate/:word

Where **word** refers to the content based on which AutoComplete event will be triggered.

- (ii) **FetchDictWord::** The API is accessed via HTTP GET method passing search word name as a parameter and returns JSON response with relevant all word details for the requested one. The entire JSON data is parsed in browser readable form at the client side and shown to the user in Grid format along with pagination for easy access.

URL :: relative_url+ /api/bengworddetails/:searchword

Where searchword refers to the word user selects from the search box of web application.

There are some important glimpses of information to be considered, such as the followings:

- [1] **Client Side Technology:** used are HTML5, CSS, JQUERY, ANGULAR JS, ANGULAR MATERIAL JS.
- [2] **Server Side Technology:** used is Spring Framework in JAVA.
- [3] **Build tool:** used in server side, to compile and fetch the reference of files and needed jars to run the entire application, is Gradle Build Tool.
- [4] **Font:** To see Bangla font without distortion we have used Gandhari Unicode font for Windows and Linux Operating System. It is also to be noted, if Google Chrome browser does not show the Bangla font in a proper manner, then it is better to change the font to SolaimanLipi or similar such Unicode compatible Bangla fonts for better and clear visibility.

4. THE DATABASE MODEL

The following database is used for storing all the information of the DBPD in an organized and normalized format. Given below is the EER diagram of the table structure (Fig. 3).

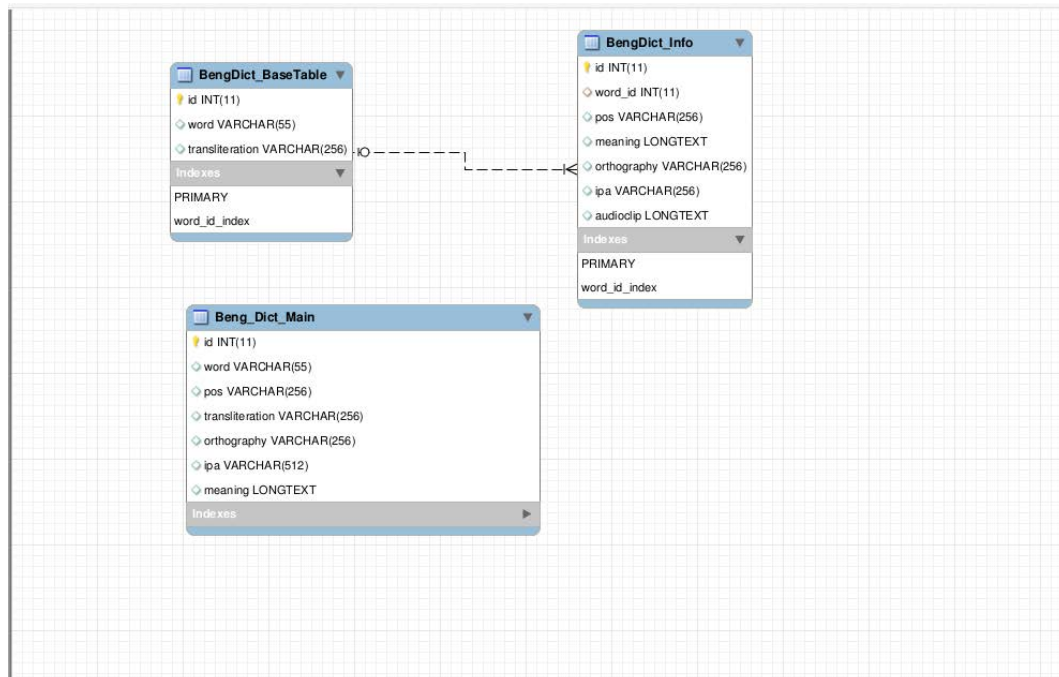


Fig. 3: EER diagram of the table structure of the DBPD

The above diagram (Fig. 3) shows the one-to-many relation between the BengDict_Basetable (Referenced table) and the BengDict_Info (Referencing Table) by establishing the foreign key constraints. The BengDict_Basetable here is called as ‘Parent Table’ and the Beng_Dict_Info is called here as ‘Child Table’. The word information in the Child Table must contain the values if the word exists in the Parent Table. The table structure and the storage procedure of the DBPD are summarized below for easy comprehension.

4.1 Table Structure Details

- (i) **Beng_Dict_Main::** This is the master table where Dictionary information is stored in a flat format. The “id” column is the auto-generated Primary Key.
- (ii) **BengDict_Basetable::** This table stores the Dictionary word names along with the transliteration information. There is one index in the table imposed on the “id” column, auto-generated Primary Key, to increase the search efficiency.
- (iii) **BengDict_Info::** This table stores all the lexicographic information like parts-of-speech (POS), meaning, orthography, IPA, audio clip related to each word in an atomic manner (there is only one value at each column field of the table), usage and the English translation. It also has word_id column used as an indexed field to increase the search efficiency. The “id” column is the autogenerated Primary Key. The word_id column of

the table creates a foreign key relation with the “id” column of BengDict_Basetable to prevent insertion of irrelevant information.

4.2 Stored Procedure Details

- (i) **LoadData_BengDict_Info::** The stored procedure rotates the word, its transliteration information from the master table (Beng_Dict_Main) to BengDict_Basetable.
- (ii) **ii>LoadData_BengDict_Info::** The stored procedure extracts each word information from the BengDict_Basetable table and fetches all the details for that particular word from the BengDict_Info table and processes the word details info like parts of speech (POS), meaning, orthography, IPA, audio clip, usage, etc. and joins the record set based on the word_id column into the BengDict_Info table.

To achieve our goal we have used the MySQL DB engine for creating our database (Digital BanglaPronunciationDictionaryDB) and have used the UTF8-UTF8_bin as a default collation for supporting the Bangla fonts in MySQL.

5. PERFORMANCE FACTOR

We have tried to add as much as Bangla words in the digital dictionary with more granular details in order to make our grammatical and semantic analysis more simplified. With this dictionary tool, we envision to get the attention from the huge audience. To make it happen, the performance is a very crucial factor to be considered. Otherwise, the entire application will become slow and will not be acceptable. There are several areas in code architecture where we have taken care of the time complexity issue. Here are the details.

5.1 Database

When we try to fetch all the lexicographic details for a particular headword and that word to be searched from a large dataset especially, both the fetching time and the duration time can play important role in the whole process. The fetched dataset is shown in the web interface for user visibility. If the fetching time from the database increases, automatically the time to display the dataset at the web interface will increase. It is in proportional relation with the database query time. As a result, the users may become impatient. In a database table, we have used the concept of indexing which will improve the search efficiency by way of generating a pointer to each record into the actual table. To achieve this we have imposed indexing on two columns:

- (a) **Word::** whenever the headword is selected by the user from the web interface that word is being searched from all the available word list in the database and gets selected the user intended word. Adding an index on word column improves this search time.
- (b) **Word_id::** When the word details to be searched from the database table, we store the word id for a particular word in a session variable and that id becomes the key to getting all information related to the the headword. We have kept separated all the information in the normalized form and tried to follow the relational model architecture. Therefore, the word details from the different database tables are fetched by the word_id column

quite quickly where the `word_id` is the primary key of the `BengDict_Basetable` table. It has been repeatedly verified that the inclusion of the indexing features to the `word_id` column increases the performance and makes search operation faster.

5.2 Auto Population

When a user starts typing the characters in the Search Word Box with the use of a virtual keyboard, all the relevant headwords get displayed in the form of the dropdown interface. This filter list is executed by an event which gets fired when the user completes typing of one character so as to reduce typing effort by displaying the word list to show what could be relevant string the user is searching for. The performance of the entire searching scheme becomes very fast and it gives users better experience reducing their waiting time.

5.3 Asynchronous Data Update

In the web application, the data is shown whenever an event will occur. There are two types of the event being fired by the user implicitly.

- (a) **Event1::** when a user starts typing, after 1 character being typed one event or action is executed, due to which the `auto_complete` list with relevant words is shown. When the event is fired, an API is called for which returns the data in the form of JSON (Javascript Object Notation). The API is a kind of protocol/interface through which data is being passed from the server end to the browser level.
- (b) **Event2::** When a user selects a word from the `auto_complete` list, an event is fired. In a similar way, it also calls for an API which returns the word details in the form of JSON. JSON is a very lightweight and self-describing data format which is primarily used here as a transport medium.

In both of the events mentioned above, the data is shown to a user at the browser level. There is a time frame difference between the events occurs and the data is displayed. The time gap is so minor (in the form of micro/nanosecond), that the end user will not feel the difference. However, the more data are stored in the database, the larger time frame gap will persist. In our application, there is no mark of any page refresh/reload. Since the entire application handles the data asynchronously, this leads to update the data only for a specific section of a page without affecting the whole one. This induces parallel way of approach in accomplishing the task which means the user does not have to wait for a long time when the data is being produced for display. Along with that, a user can also check/do other things in that corresponding page. This way is named as AJAX function which allows web pages to be updated partially by exchanging small amounts of data with the server at the backend and helps to reduce unnecessary waiting time and provides a better way of abstraction by handling call-back function and customizing the response output.

6. DEVICE INDEPENDENCE

The Digital Bangla Pronunciation Dictionary (DBPD) is designed to reach up to as many users as possible so that people from anywhere, any type of computer, mobile, and similar

other devices could access the dictionary. The architecture is created with Angular Material Design. It has an attribute named as 'flex' which could be used to handle mobile design, Tab design and desktop design in parallel. It is also interesting to note that the Code Resources could be managed from within one single repository which may save time and effort to a significant level. This implies that there is no separate code version maintained for mobile or tab. From only one code level version we could easily manage three device designs based on the need of the dictionary users. The code design also helps us make the web pages being more responsive and dynamic. Given below are the designs of the dictionary tool for mobile and tab device (Fig. 4ab):

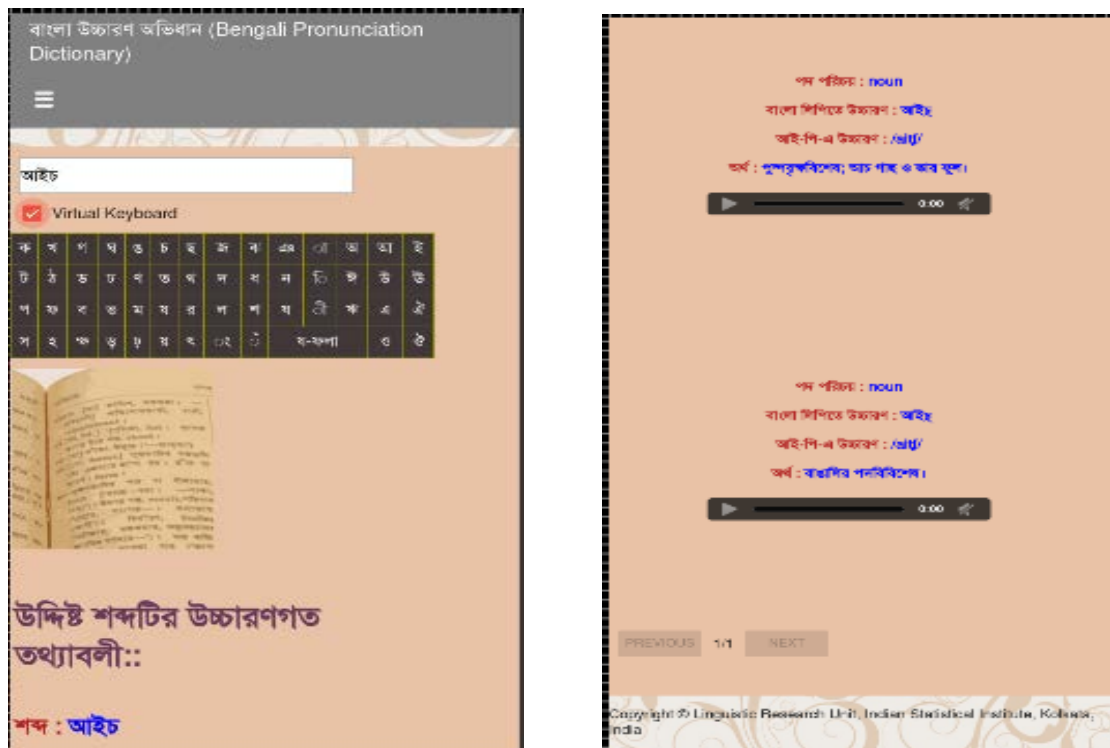


Fig. 4: Design of the dictionary tool for mobile (Fig. 4a) and tab device (Fig. 4b):

7. THE BENEFICIARIES

With the direct utilization of the DBPD discussed above the learners will learn how the Bangla words should be pronounced in standard or acceptable manner. In the era of on-line education also a digital resource of this kind has the potential to improve on the traditional methods of language teaching where the learners can avail opportunities to learn the standard or acceptable pronunciation of words with the direct utilization of modern computer technology in an interactive fashion with the constructive assistance of language teachers.

Through activation of a dialogue-based interactive user interface, learners will know how the words in Bangla are pronounced in an acceptable manner when the orthography of words hardly matches with their pronunciation. This dictionary will also provide opportunities to the learners to understand how variations in pronunciation are caused due to variations in part-of-

speech and meaning of words. Within the scheme of on-line education, the dictionary proposed here can be equally useful for the non-native and foreign learners as well as for the speakers of different Bangla dialects and regional varieties as the speakers will get good opportunities and exposure to learning the particular pronunciations of words which are considered standard and acceptable in the language.

This DBPD is, perhaps, the first of its kind in Bangla and other Indian languages. It is fully computer-assisted with multimedia interface facilities with a regular scope for data up-gradation, information augmentation, and system modification (Dash 2011). It has also facilities for speech output, which can be effectively used in teaching Bangla as the first or the second language, on-line language teaching, text to speech conversion, language recognition, word recognition, machine learning, machine-aided translation, lexicography, word-sense disambiguation, and E-governance. It can also be used to train the linguistically impaired people in recognition and production of standard Bangla speech. The immediate beneficiaries of this dictionary are the native Bangla learners, foreign Bangla learners, Bangla teachers, text to speech system developers, translators, system designers, computational lexicographers, language planners, speech pathologists and descriptive linguists and the cognitive linguists. Perhaps the common Bangla speakers are the most notable beneficiaries.

8. CONCLUSION

In the present era of information technology, a digital pronunciation dictionary may be considered as an indispensable resource in Computer Assisted Language Learning (Felix 2008, Leakey 2011), Blended Learning (Garrison and Kanuka 2004, Garrison and Vaughan 2012), and Data Driven Learning (Kosem 2008, Boulton 2009, Boulton 2010, Boulton 2011). Language teachers and course designers can often incorporate this resource into an on-line and/or off-line language education system, where teaching learners about the pronunciation of words is an important part of the system (Jones 1986, Warschauer and Healey 1998, Bax 2003). Such application potentials have acted as a motivation behind the development of the resource for the Bangla language.

This DBPD, if compared with the pronunciation dictionaries available in printed and digital format in Indian and foreign languages, invariably differs in its form, content, treatment and interface as depicted above. However, the striking deficiency of the present DBPD is that it does not relate with or refer to the pronunciation of Bangla words used in Bangladesh because it has not been possible for us to get access to the particular pronunciation type used in this country. If we can gather an adequate amount of information of pronunciation used in Bangladesh, we shall be more than happy to add up that part in this dictionary to make this resource more useful across the borders.

Acknowledgement:

This is a thoroughly revised paper presented at the 22nd *Himalayan Languages Symposium (HLS-22)*, Indian Institute of Technology, Guwahati, Assam, India 08-10 June 2016.



References

- Bax, S. (2003) Computer Assisted Language Learning (CALL): Past, Present, and Future. *System*, 31(1): 13-28.
- Boulton, A. (2009) Testing the limits of data-driven learning: Language proficiency and training. *ReCALL*, 21(1): 37-51.
- Boulton, A. (2010) Data-driven learning: On paper, in practice. In: T. Harris & M. Moreno Jaén (eds.) *Corpus linguistics in language teaching*. Bern: Peter Lang. Pp. 17-52.
- Boulton, A. (2011) Data-driven learning: The perpetual enigma. In: S. Goźdz-Roszkowski (ed.) *Explorations across languages and corpora*. Frankfurt: Peter Lang. Pp. 563-580.
- Dash, N.S. (2011) Some physical advantages of an electronic dictionary. *Indian Linguistics*. 71(1-4): 93-102.
- Dash, N.S. (2013) Developing a Digital Pronunciation Dictionary in Bangla for Computer-Assisted Language Teaching, E-Learning, and Language Technology. Presented at the 35th International Conference of the Linguistic Society of India (ICOLSI-35), Central Institute of Indian Languages, Mysore, India, 27-29 November 2013.
- Felix, U. (2008) The unreasonable effectiveness of CALL: what have we learned in two decades of research?. *ReCALL*. 20 (2): 141–161.
- Garrison, D.R. and H. Kanuka (2004) Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*. 7(1): 95–105.
- Garrison, D.R., and N.D. Vaughan (2012) *The Future: Blended Learning in Higher Education: Framework, Principles, and Guidelines*. Published Online: 5 Jan 2012
- Jones, G. (1986) Computer simulations in language teaching-the KINGDOM experiment. *System*. 14(2): 179-186.
- Kosem, I. (2008) User-friendly corpus tools for language teaching and learning. In: A. Frankenberg Garcia (ed.) *Proceedings of the 8th teaching and language corpora conference*. Pp. 183-192. Lisbon: ISLA-Lisboa.
- Leakey J. (2011) *Evaluating Computer Assisted Language Learning: an integrated approach to effectiveness research in CALL*, Bern: Peter Lang.
- Warschauer, M. and D. Healey (1998) Computers and language learning: an overview. *Language Teaching*, 31(1): 57-71.

Websites:

- [1] <https://www.mysql.com/>
- [2] <https://material.angularjs.org/latest/>,
- [3] <http://www.w3schools.com/angular/>,
- [4] <https://scotch.io/tutorials/angular-routing-using-ui-router>
- [5] <https://spring.io/>,
- [6] http://www.tutorialspoint.com/spring/spring_architecture.htm
- [7] https://en.wikipedia.org/wiki/Single-page_application,
- [8] <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>

