A Framework for Building a Natural Language Interface for Bangla

Yeasin Ar Rahman, Mahtabul Alam Sohan, Khalid Ibn Zinnah, Mohammed Moshiul Hoque
Chittagong University of Engineering & Technology
Chittagong-4349, Bangladesh
nibir201188@gmail.com, mahtabul1993@gmail.com, khalidex@yahoo.com, moshiul_240@cuet.ac.bd

***Abstract*—** Mobile Computing Devices are enabling connection between people and the Internet, largest source of information in the world. However to properly utilize this knowledge in these devices most people need a Natural Language Interface. Siri, Google Now, Cortana are examples of such interfaces. Because Bangla is a low resource language building such interface is very difficult and time consuming. However due to the increasing numbers of smart phone and smart device users in the Bangla speaking regions, application developers are facing the need for such interfaces to provide web services effectively. This paper addresses this issue and gives an empirical framework on how to build a feasible Natural Language Interface in Bangla and similar low resource languages.

Keywords*— *Bangla Natural Language Interface; Human Computer Interaction; Bangla Speech to Text; Bangla Language Processing; Artifical Intelligence.

I. INTRODUCTION

Language is used to communication. There are several methods for communicating. They are verbal, written and visual (sign language, body movement, nod, gesture etc.). Each of the form of communication is very unique that it is very difficult for computers to understand the meaning. However after the introduction of smart mobile devices such as smart phones, wearable devices such as smart watch, smart band etc. and Internet of things traditional user interface such as GUI (graphical user interface) and CLI (command line interface) are no longer a viable option for effective use of human time [1]. In many cases these devices don't have traditional input-output devices such as keyboard, mouse or display. In order to give pleasant user experience Natural Language Interface (NLI) or generally Voice User Interface (VUI) are very practical and somewhat necessary approach in these devices. NLI generally has an artificial intelligence unit so it can successfully distinguish between different commands of the user and then confidently perform the task which the user intended. The process utilizes Automated Speech Recognition and Natural Language Understanding. Generally this kind of artificial intelligence program or service is called an Intelligent Personal Assistant (or simply IPA) which can perform tasks or services for a user [2]. In a NLI linguistic events such as verbs, phrases and clauses act as User Interface (UI) controls for searching, creating, selecting and modifying data in software applications. In interface design NLIs are sought after for their speed and ease of use, but most suffer the challenges to understanding wide varieties of ambiguous input [5]. Processing a language consists of some important steps. They are sequentially Morphological, Syntax and Semantic analysis. For a Natural Language Interface Part of Speech Tagging, Named Entity Recognition, Intent analysis are important fields for achieving desired result. Using machine learning methods the state of the art systems has achieved striking results [3]. However one of the key reason for these success were large amount of annotated data. For most of the languages in the world such data is not available.

Over 250 million people use Bangla as their medium of communication. Currently there are 60 million internet users in Bangladesh [23]. Most of these users use internet through mobile devices. Currently web application developers and mobile app developers do not have access to natural language processing technologies such automatic speech recognition, text to speech, natural language searching, optical character recognition etc. in Bangla. Since most of the general population in Bangla speaking regions are not proficient in English it is not possible to provide satisfactory service through mobile devices without these natural language services. On the other hand most of the applications and services currently available in this region cannot be fine-tuned for the regional preferences without these natural language services. Developing a Natural Language Interface is important for Bangla because it facilitates easier search for information, enables automation of home and industry, makes communication with robots possible, enables navigation better and easier, allows people with disabilities to easily communicate with people and devices. Most importantly it removes the language barrier to enter internet and thus allows the inclusion of the population in villages to take part in the digital world. Through a robust NLI it is possible to effectively provide digital services to rural and urban population. There is very little data available for Bangla Language Research. So it is very difficult to develop a Natural Language Interface.

In this work we address this issue and provide a standardized guideline and an empirical framework for building a Bangla natural language interface. Although there is a lack of available data, we believe if we effectively use the data available in Bangla in the internet, it is possible to build a feasible system which can provide natural language services to application

978-1-5090-5627-9/17/$31.00 ©2017 IEEE

developer for providing intelligent services to the Bangla speaking population.

II. BACKGROUND

There have been significant research in the field and subfields (e.g. Human Computer Interaction) of artificial intelligence to build a system that can interact with human at the same level of understanding of another human. But research suggest that it is quite difficult to formulate a methodology for simulating human behavior in machine as the knowledge of understanding human behavior is still at primitive level. Practical systems are designed to perform specific tasks in specific fields.

The open agent architecture (OAA) [21] is one of the most prominent framework for building multimodal interface. It delegates most of the work to individual services and thus allows a clean design. It is domain independent framework. However it requires that such services are available. But in most of the low resource languages those services do not exist and has to be integrated into the system.

RADAR [22] is another notable system that allows to maintain a calendar intuitively. It was a personal assistant type system. It is task specific framework. However it paved the way modern personal assistant architecture.

However in the domain of intelligent personal assistant the most influential project in terms of technology was the CALO project [11] which was funded by DARPA (Defense Advanced Research Projects Agency). One of the products of this project was PAL (Personal Assistant that Learns). It provided a general guideline for building useful NLI system. These are the key aspects of the system such as learning, data management, data acquisition, controllers and user Interface.

Today's most prominent Intelligent Personal Assistants for example Siri(Apple), Cortana(Microsoft), Watson(IBM), M(Facebook), S voice(Samsung), Google Now(Google) etc. all use similar concepts for their NLI platforms. But all these systems were made for English speaking users primarily. To our knowledge there has not been any significant work to build an NLI in Bangla. However most of the science for constructing such an interface in Bangla has been explored by the researchers. Here significant works for Bengali Speech Recognition and knowledge representation and reasoning is stated.

In almost all the cases NLI utilizes a speech recognition engine. The first large vocabulary continuous speech recognition system was Sphinx-II [4] developed in the Carnegie Melon University. In most of the modern general-purpose speech recognition systems Hidden Markov Model is used. HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. However the state of the art technology in speech recognition is Deep Neural Network. Using very large amount of data researchers from Microsoft, Google, IBM, Baidu, Apple, Nuance etc. companies have reached near perfect accuracy [8]. DNN architectures generate compositional models, where extra layers enable composition of features from lower layers, giving a huge learning capacity and thus the potential of modeling complex patterns of speech data [9]. In case of Bangla there exists no general purpose speech recognition service. There exist no online or offline engine to covert There have been several works on Bengali speech to text systems, almost all of the studies generally emphasized on developing new algorithms rather than the implementation of an Application or Service. Hasnat et al. [12] made a customized Hidden Markov Model (HMM) based scheme for pattern classification. They also integrated the stochastic model within the scheme for Bengali speech–to-text. They used the HTK framework to make their test system. Firoze et al. [13] developed a fuzzy logic based speech recognition system and proposed that fuzzy logic has to be the base for all linguistic ambiguity-related problems in Bengali. They empirically showed that fuzzy logic results in improved response for more ambiguous linguistic entities in Bengali speech. This study is the first attempt using cepstral analysis in the artificial neural network (ANN) to recognize Bengali speech. The only large vocabulary Bengali continuous speech recognition system to our knowledge is Shruti-II developed in IIT, Kharagpur for visually impaired person [14]. Speech to text (STT) engine outputs textual output in formats specified by the developers. It can be further processed by other application processes.

Natural Language Understanding part of a NLI makes use of Named Entity Recognition (NER), Parts of Speech Tagging (POS Tagger), Bangla Wordnet and sentence parsing. For parts of speech tagging Maximum Entropy (ME) approach was proposed by Ekbal Asif [15]. In 2007 Hasan et el. performed a comparative study on HMM, Unigram and Brill's method and showed that brill's methods gives better performance [16]. Named Entity Recognition is very difficult for low resource languages like Bangla. Asif Ekbal proposed a method using support vector machine for Bangla NER in 2008[17]. Cucerzan, Silviu, and David Yarowsky showed a method for language independent named entity recognition [18]. For NLI dependency parsing is quite useful. Das, Arjun, and Arabinda Shee Utpal Garain evaluated two different dependency parsers [19]. Sankar et el. used demand satisfaction approach for dependency parsing in Bangla [20].

Most of the previous work in Bangla has one key limitation that is almost all of them were tested on a small dataset which is not suitable for a NLI.

III. METHODS

A NLI is a complex system. So the internal organization is divided into several parts or modules. The proposed system has the following modules Automatic Speech Recognition Unit (ASRU), Intelligent Response Unit (IRU), Knowledge Base, Question-Answering and Search Engine, Control Systems API, Text to Speech based response system. The system architecture is shown in Fig. 1. Each module in the system is important for the system to work properly. However the internal working of individual module is separate from the architecture. So the each module can be changed and improved without interfering with other components. The approaches used in different modules to perform its task is given in the following sections.

A. Automatic Speech Recognition Unit

The primary user interface of our system is voice based, so there is a great emphasis on voice input. The unit works the following way the user gives a command or query to the device (computer/mobile). Then the system takes that voice data and decodes it using CMU Sphinx toolkit. CMU Sphinx [10] toolkit is a HMM based large vocabulary continuous speech recognition system that supports any language if the Acoustic

Model and Language Model are provided. Here CMU Sphinx is used because the required acoustic model and language models are easier to train for this toolkit. Also there is readily available helping applications which are necessary to train the models. After decoding the speech signal CMU Sphinx gives a text output that is then sent to IRU. The IRU then uses various natural language processing techniques to evaluate the sentence to correctly identify the intent of the user.

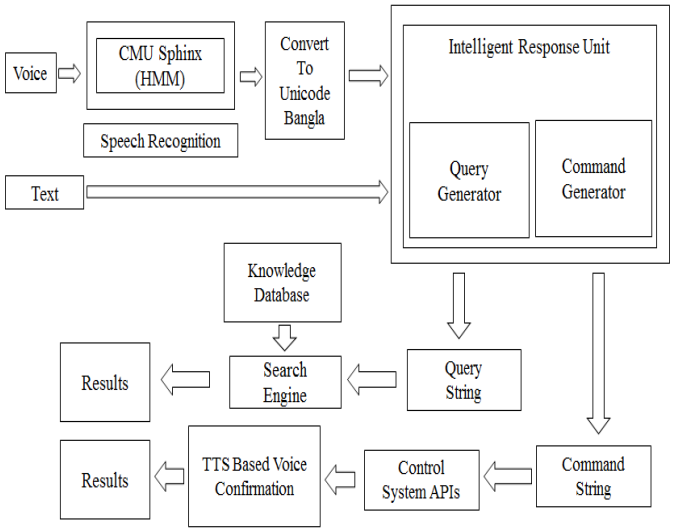

Fig.1 *System Architecture*

B. *Intelligent Response Unit*

Intelligent Response Unit (IRU) is the most important component of the proposed system. The responsibility of IRU is to transform human language into actionable data. It uses several natural language processes. These components are implemented using Apache OpenNLP [24] framework. Each process is a separate component inside the system. The architecture for IRU is shown in Fig. 2.

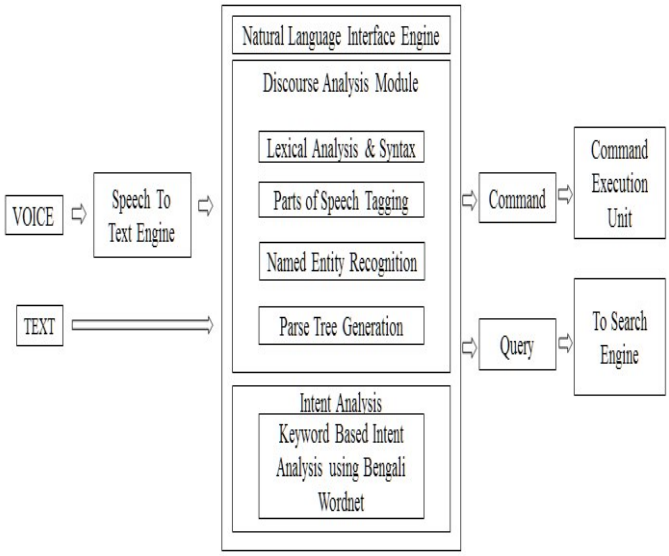

Fig. 2 *Intelligent Response Unit Architecture*

A brief explanation about the components working procedure is given below:

1) Lexical Analysis and Stemming: Takes textual input from the previous units. Convert the input into tokens using lexical analyzer which stems individual words to roots word or converts sentence into word tokens. Each token is then passed to the next section for example,

Original Form: "বিরানী কোথায় সবচেয়ে ভাল?"

Tokenized form: "বিরানী", "কোথায়", "সবচেয়ে", "ভাল", "?"

2) Parts of Speech Tagging: The parts of speech(POS) tagger tags each token into a parts of speech tag. It is a very important step because all the steps afterwords greatly depends on correct tagging of POS. It uses statistical MaxEnt based learning model. OpenNLP is used to train the model. Here (1) is a simple form for calculating MaxEnt.

$$p(c|x) = exp\frac{\exp(\sum_{i=0}^{n} Wcifi(c,x))}{\sum_{c\prime\epsilon C} exp(\sum_{i=0}^{n} Wc\prime ifi(c\prime,x))} \quad (1)$$

Here the symbols represent their universal meaning.

3) Named Entity Recognition: The named entity recognizer will recognize all the named entities in the text. The named entities are the objects that the system will run a command on or search for an answer. It also uses MaxEnt model. For example,

Tokenized form: "বিরানী", "কোথায়", "সবচেয়ে", "ভাল", "?"

Named Entity: : <Entity Food>"বিরানী"</Entity Food>, "কোথায়", "সবচেয়ে", "ভাল", "?"

4) Sentence Dependency Parsing: The dependency parser creates a tree structure of the text that defines the relationship between different parts of the sentence. It is used for the command or query output generation. It requires a tree-bank. The dependency parsed tree for the sentence "কাজী নজরুল ইসলাম কোথায় জন্মগ্রহন করেছেন?" has been given in Fig. 3. The figure is the output from the annotation application webanno[26].

Fig. 3 *Example of dependency parsing*

5) Keyword Based Intent Analysis: The intent analysis is based on keyword. Each keyword can be rephrased by other words given the word is very close to the word in Bangla wordnet. Here a sample of process is shown in Table I.

TABLE I. INTENT ANALYSIS

Intent	Keyword	Alternate Keywords
Weather	আবহাওয়া	পরিবেশ, অবস্থা
Control	জ্বালাও	অন করো

An intent is dependent on the application service supported by the application developer. Intents can have multiple keywords for different processes. If two intent keyword collide then the named entities are used to resolve the collision.

6) Output Generation: After the intents has been recognized. The system creates a output string based on the

parsed text and entities recognized. Here a sample is given in Table II.

TABLE II. OUTPUT GENERATION

Utterance	Output
Type: Command "এসি টেম্পারেচার ২৫ এ দাও"	{intent : AC control command: Set Temp value : 25 degree place : this.room type : command}
Type: Query "আজকে রাতের পরিবেশ কেমন হবে?"	{intent : weather day : this.today.date place : this.location time : "রাত" type : query}

C. Knowledge Base

Knowledge base is the physical database where all the information remains. The system will use Apache Solr [25] Based knowledge database, which is Open Source Full Text Search Engine Database. The data sources are following

1) Wikipedia
2) Banglapedia
3) Newspaper, Blog and Bangla Website Crawl Data

D. Question Answering and Search

For Factoid question answering (QA) the system will use the general methodologies for question answering system. It uses TF/IDF [7] based algorithm for ranking the necessary documents. The weight of term i in document j can be found from the document term matrix using (2). We use a cut-off number of top 20 result to keep the calculation time minimum.

$$w_{i,j} = tf_{i,j} X\, idf_i \quad (2)$$

After finding the candidate passage. The QA systems task is following:

1) Question Classification: It analyzes the question using Webclopedia QA Typology [6]. All the factoid question follow a pattern. So using question classifier can understand the type of possible result.

2) Answer Type Pattern Extraction: After classifying the question the system consolidates candidate sentences. It uses the following independent methodologies [7] to rank the candidate sentences.

a) Answer type match
b) Pattern match
c) Number of matched question keywords
d) Keyword distance
e) Novelty factor
f) Apposition features
g) Punctuation location
h) Sequences of question terms

If the confidence score is more than 70% for an answer that answer is shown. Otherwise top five answers and their associated document link is displayed. In case of open-domain question the system leverage it to the search engine. We use the above mentioned Apache Solr Search System.

E. Control Systems APIs

The Control System APIs are set of methods those are exposed in order to control a device or application programmatically. Each system has different set of control API. Any API can be logically combined to our proposed framework. The APIs takes the command output from the IRU and then performs a task specified in the command. Applications developers has to utilize the intents used in the previous steps in order to design the APIs.

F. Text to Speech Confirmation

After completing all the steps it is necessary to confirm the user that the task has been completed. The task can be accomplished by Text to Speech (TTS) engine. In the cases where the results are output of a query the results can be directly shown in the display. If no display device is present then it can be output by TTS engine.

IV. EXPERIMENTS

This section describes the process of data collection, data collection environment, tools used for collecting data and primary results.

A. Data Collection

For the purpose of robust speech to text conversion we tried to find out the possible utterance a user might say during an operation. As there has not been any previous attempt to make a NLI for Bangla, we constructed a Command Script for Bangla adapting similar utterances as used in other languages. The script has 250 utterances. The Domains covered in the script is given in Table III. We also found usually for most command and control utterance the vocabulary size is limited. However precision of this kind of command is more important because the commands are in many cases similar and it is possible that one command might be confused with another one. The solution to this problem is to collect large amount of data to capture accurate voice properties of the users. The voices were taken from the users in the age range of 20-30 years. Data was taken mainly from the people who volunteered for the project. The data is roughly divided into 75% male and 25% female ratio. The occupation of the users are mostly students with some exception of service holders and businessmen. The education level of the volunteers vary from higher secondary to university. The voice was taken in the standard dialect of Bangla. It has been tried to minimize regional bias by taking voices from people of different districts.

TABLE III. COMMAND AND SERVICE SCRIPT

Domain Name	Example Utterance
Weather	আজকের তাপমাত্রা কত?
Food And Restaurant	বিরানী কোথায় সবচেয়ে ভাল?
General Direction for Items (Stationary/Grocery)	আশেপাশে বইখাতা কোথায় পাওয়া যাবে?
Directions for Landmarks	বনানীতে হাসপাতাল কোথায়?
Travel Related	কাছাকাছি দেখার মত কি আছে?

Price Related (General/Stock Market)	গতকাল [মুরগীর] দাম কত ছিল?
General Query	বিরানির রেসিপী বের কর৷
General Knowledge Question	কাজী নজরুল ইসলাম কোথায় জন্মগ্রহন করেছেন?
Note and Alarm	সকাল ৭ টায় এলার্ম সেট কর৷
Communication	বাসায় ফোন দাও
Conversion and Calculation	৪৫ ডলার সমান কত টাকা?
Sports Query	বাংলাদেশের স্কোর কত?
Transportation Timetable	আজকে পারাবত লঞ্চ কয়টায় ছাড়বে?
Device Control	এসি টেম্পারেচার ২৫ এ দাও
Maps and Fair Calculation	ধানমন্ডি ২৭ থেকে ধানমন্ডি ১০ এর রিক্সা ভাড়া কত?
Numbers	এক হাজার দুইশ পাচ
Program Control	আমার কল রেকর্ড দেখাও

On the other hand in case of queries it is not practically possible to guess what a user might say. To address this issue we created another script which constitutes of various text collected from newspapers, television, novel and other domains which contains common words in everyday use. A partial list of data Sources in our query script is given in Table IV.

The voice data has been collected under the following conditions:

1) Microphone: A Blue Yeti Professiona microphone is used to collect the data

2) Software: To collect the data Audacity software is used.

3) Environment: The data has been tried to collect in a natural environment in order to take the noise in the natural environment into consideration.

TABLE IV. DOMAINS FOR GENERAL QUERY SCRIPT

Source Name	Percentage of total utterance (Rounded)
Wikipedia	20%
Newspaper Editorial (Prothom-Alo)	15%
Blogs (Somewherein Blog, Bdnews24 Blog)	20%
Websites (techtunes, techtweets)	10%
News (Prothom-Alo, Bdnews24)	10%
Novels (Dorojar Opashe, Parapar, Meku Kahini)	10%
History (Ekattorer Dinguli)	5%
Famous Personalities (Humayun Ahmed)	5%
Famous Places (Cox's Bazar, Jaflong)	5%

For the Knowledge database all the articles from Bangla Wikipedia has been collected and hand cleaned. It contains 240K lines, which to our knowledge is one of the largest Bangla Text Corpus. Data from popular newspapers, blogs and websites are being collected. As most of the open source nlp software do not directly support Bangla in Unicode form. We needed a transliteration tool to easily port Bangla support in these software. Open-source transliteration tools for Bangla are not suitable for large amount of text as they are very slow. We developed a Fork-Join Framework based parallel processing transliteration application which speed-up our process several times.

B. Results

The system development is in the preliminary stage. Currently the data collection phase is going on. Several Hours of Voice data has been collected, it is being tested to see if the quality matches the expectation. The data collection is on process. Here the result found for first 2 hours of command and service script is given. It has been tested for two different models a semi-continuous model and a continuous model. Here two parameters has been used to test the data. The word error rate (WER) is the number of wrongly recognized words in every 100 words recognized, it measures accuracy of the system. On the other hand response time (RT) measures how much time the system takes to recognize a single sentence. It has been found that semi-continuous models are faster in detecting utterance and continuous models are more accurate. Here accuracy and rounded response time is given in Table V. After the data collection is complete this accuracy can change.

TABLE V. WORD ERROR RATE AND RESPONSE TIME

Model	WER	RT
Continuous (800 words)	6.778	0.8s
Semi-Continuous (800 words)	8.598	0.5s

V. CONCLUSION

The primary contribution of this work is formulating a feasible framework for low resource languages like Bangla. In this work previously done research has been proposed to implement a particular feature of the system and in cases where no such work is available it has been suggested to adapt the existing systems into Bangla. The Secondary contribution of this work is the construction and design of large datasets for different natural language processes. Our planned Bangla Speech Corpus is the largest speech corpus yet designed for Bangla as far as we know. The language model is also very large to easily use in any large scale production environment. In continuation of this work the experimental result and accuracy data will be published. The performance analysis and hardware requirements for specific tasks will be mentioned. Further research is needed for a machine learning based Question Answering engine. Also the Deep Learning can be used for speech recognition engine.

REFERENCES

[1] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, no. 7, pp. 30–40, Jul. 1994.

[2] N. R. Jennings and M. Wooldridge, "Applications of Intelligent Agents," *Agent Technology*, pp. 3–28, 1998.

[3] S. J. Russell and P. Norvig, "*Artificial intelligence: A modern approach*," 3rd ed., Prentice Hall, pp. 25-28 2009.

[4] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech & Language*, vol. 7, no. 2, pp. 137–148, 1993.

[5] P. R. Cohen, "The role of natural language in a multimodal interface," *Proceedings of the 5th annual ACM symposium on User interface software and technology - UIST '92*, 1992.

[6] D. Ravichandran and E. Hovy. "Learning surface text patterns for a question answering system." In *Proceedings of the 40th annual meeting on association for computational linguistics,* Association for Computational Linguistics, pp. 41-47, 2002.

[7] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition," 2nd ed. United States: Pearson Prentice Hall, Ch. 23, pp. 5 -23 , 2008.

[8] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of Four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[9] L. Deng and D. Yu, *Deep learning: Methods and applications*. Grand Rapids, MI, United States: now publishers, pp. 198–213, 2014.

[10] P. Lamere *et al*., "Design of the CMU sphinx-4 decoder." In INTERSPEECH. 2003.

[11] P. M. Berry, K. Myers, T. E. Uribe, and N. Y. Smith. "Constraint solving experience with the calo project" In *Changes' 05 International Workshop on Constraint Solving under Change and Uncertainty*. 2005.

[12] M. A. Hasnat, J. Mowla, and M. Khan, "Isolated and continuous bangla speech recognition: implementation, performance and application perspective," in Center for research on Bangla language processing (CRBLP), 2007.

[13] A. Firoze, M. S. Arifin, and R. M. Rahman, "Bangla user Adaptive word speech recognition," *International Journal of Fuzzy System Applications*, vol. 3, no. 3, pp. 1–36, 2013.

[14] S. Mandal, B. Das, and P. Mitra. "Shruti-II: A vernacular speech recognition system in Bengali and an application for visually impaired community." *Students' Technology Symposium*, IEEE, 2010.

[15] A. Ekbal, R. Haque, and S. Bandyopadhyay. "Maximum Entropy Based Bengali Part of Speech Tagging." A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal 33: 67-78, 2008.

[16] F. M. Hasan, N. UzZaman, and M. Khan. "Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla." *Advances and Innovations in Systems, Computing Sciences and Software Engineering*. Springer Netherlands, 121-126, 2007.

[17] A. Ekbal, and S. Bandyopadhyay. "Bengali Named Entity Recognition Using Support Vector Machine." *IJCNLP*. 2008.

[18] S. Cucerzan, and D. Yarowsky. "Language independent named entity recognition combining morphological and contextual evidence." *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*. 1999.

[19] A. Das, and A. S. U. Garain. "Evaluation of two Bengali dependency parsers." 24th *International Conference on Computational Linguistics*. 2012.

[20] S. De, A. Dhar, and U. Garain. "Structure Simplification and Demand Satisfaction Approach to Dependency Parsing for Bangla." *Proc. of 6th Int. Conf. on Natural Language Processing (ICON) tool contest: Indian Language Dependency Parsing*. 2009.

[21] A. Cheyer, and D. Martin. "The open agent architecture." *Autonomous Agents and Multi-Agent Systems* 4, no. 1: 143-148, 2001.

[22] P. J. Modi, M. Veloso, S. F. Smith, and J. Oh. "Cmradar: A personal assistant agent for calendar management." In *Agent-Oriented Information Systems II*, pp. 169-181. Springer Berlin Heidelberg, 2005.

[23] B. T. R. Commission, "Internet subscribers in Bangladesh July, 2016," 2016. [Online]. Available: http://www.btrc.gov.bd/content/internet-subscribers-bangladesh-july-2016. Accessed: Sep. 1, 2016.

[24] T. A. S. Foundation, "Apache OpenNLP," 2010. [Online]. Available: https://opennlp.apache.org/. Accessed: Sep. 6, 2016.

[25] T. A. S. Foundation, "Apache Solr," 2016. [Online]. Available: http://lucene.apache.org/solr/. Accessed: Sep. 6, 2016.

[26] R. E. d. Castilho, C. Biemann, I. Gurevych and S.M. Yimam, "WebAnno: a flexible, web-based annotation tool for CLARIN". *Proceedings of the CLARIN Annual Conference (CAC),* 2014.