An Intelligent Bangla Conversational Agent: TUNI

Chapter · May 2021

DOI: 10.1007/978-981-16-0586-4_34

3 authors:

Md Tareq Rahman Joy
Khulna University of Engineering and Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Md. Nasib Shahriar Akash
Khulna University of Engineering and Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

K. M. Azharul Hasan
Khulna University of Engineering and Technology
96 PUBLICATIONS **472** CITATIONS
SEE PROFILE

An Intelligent Bangla Conversational Agent: TUNI

MD Tareq Rahman Joy, Md. Nasib Shahriar Akash and K. M. Azharul Hasan

Department of Computer Science and Engineering
Khulna University of Engineering & Technology, Khulna-9203, Bangladesh
me.tareqjoy@gmail.com, akashshahriar55@gmail.com,
az@cse.kuet.ac.bd

Abstract. This research paper represents a method of a man-machine communication system named "TUNI", where TUNI acts like a psychologist and continuously asks questions based on the response of the user. It is important in Bangla Natural Language Processing as it processes Bangla text and responses like a human. In this paper, a new rule-based method is used to extract the input information and to map them with output. After preprocessing the input text, it is checked with the rules of the corpus. If a rule is matched, then one corresponding response rule is selected from the corpus. Then the response rule is processed and given to the user as output using the user interface. In this method, most of the time, TUNI produces syntactically correct responses.

Keywords: Corpus, POS-Tag, Tokenization, Stemming, Sentiment analysis, Decomposition rule, Reassembly rule

1 Introduction

A Conversational Agent [1]-[2] is the way to interact with a machine using natural language like English, Bangla, etc. Eliza was the first natural language conversation program described by Joseph Weizenbaum in January 1966 [3]. From then, several natural language conversation programs are implemented in the English language which may be in different contexts. But for Bangla language, there aren't any remarkable analysis and research for a man and machine communication system. But, after one of the popular language - English, Bangla is the fifth most popular language [4], which is the state language of Bangladesh. So, we have proposed a corpus-based method [5] which establishes a natural language communication system between man and machine in Bangla language. The machine we have implemented, TUNI, imitates and responses like a psychologist who can communicate with a user through a comfortable terminal-like interface. Though the machine can mimic like a mock psychologist, the editing capabilities of TUNI "corpus" gives the ability to respond in various context.

Bangla language has more complexities than any other language [6] because the pattern of Bangla language isn't fixed like other languages as an example – English. Suffixes vary depending on the grammatical person. Even there is a different form of second-person which also causes suffix change. So, the simple approach can't be

enough for Bangla Conversational Agent. We have tried with a new approach, pattern matching with Bangla POS tagger [7] and very basic sentiment analysis. With this approach, TUNI can produce natural and syntactically correct replies overcoming those complexities.
By this method, a corpus is made to match with the input text and respond according to the context. There are rules many rules available in the corpus that are used to match and extract the information from the input, called decomposition rules. A preprocessing is necessary for making the input text suitable for the decomposition rules. The preprocessing step normalizes the input text as well as adds POS tags to normalized text. For each decomposition rule, there are some rules which use the parsed information and make ready for the response, called reassembly rules. Using this method, our conversational agent performs well compared to other knowledge-based methods in Bangla. As it is handwritten rules, the responses are relevant to the context and accurate.
Further, the paper is organized into sections like this: Section 2 describes the related work in this field, Section 3 describes the additional tasks that are needed for our proposed method, Section 4 describes the proposed method for our system, Section 5 illustrates the performance of our proposed method, Section 6 concludes all the section of the paper.

2 Related Works

ELIZA was a domain-based system that uses some semantic rules that simply matches the keyword from an extensive collection of rules. And then decompose the input string to remove the semantic conflicts. At last, it navigates through several context-based replies to find a suitable reply to the corresponding input. It was developed using MAD-Slip language that provides a vast collection of data structure that helps in string manipulation. It was based on a SCRIPT, a collection of rules to find the focus of an input given by the user and analyze the semantic conflicts. The editing ability of the SCRIPT gives the system capability of changing the context.
ALICE (Artificial Linguistic Internet Computer Entity) was invented by Richard Wallace in 1995, is a knowledge-based natural language processing system that can communicate with a human by processing the user input using some reasoning heuristically pattern matching rules [8]. It was build based on XML Knowledge bases [9]. It matches with some predefined response using a conversational agent that is developed based on artificial intelligence.
PARRY was discussed by psychiatrist Kenneth Colby, at Stanford University in 1972 [10]. It used to simulate a paranoid schizophrenia patient. It embodied a conversational strategy, and as such it was a program far more serious and advanced than ELIZA. It was described as "attitude ELIZA" In the early 1970s PARRY was tested using a variation of the Turing Test. A group of experienced psychiatrists analyzed a combination of real patients and computers running PARRY through teleprinters.

DOLY [11] is a man and machine communication system that was implemented to give replies to a user query in a human manner for the education system in the Bengali Language. It is an AI-based human-computer interaction system where machine learning algorithms are used with the help of Bengali Natural Language Processing (BNLP). It responds to the answer to a question given by a user using the collection of knowledge and finds the desired output. There is a train function adapter that trains the DOLY using the knowledge of replies.

3 Prerequisite Works

For our proposed method, there is a need for some additional concepts. They are described below:

3.1 Bangla Pos Tagging

Though POS tagging is the heart of our proposed method, only a few POS tags are required to detect the pattern of a sentence. Generally, noun, pronoun, verb and adjective are enough for detecting the pattern of a sentence. Additionally, the different form of pronoun and verb are also detected for getting the proper sense of the grammatical person. For adding POS Tag to each word of a sentence, a dataset is used. The dataset contains the direct mapping of each word with its POS tag. POS tags are described in Table 1.

Table 1. Example of POS tag

POS Tag	Description	Example word
adj	Adjective	বড়
p.1	Pronoun – first person	আমার
p.2	Pronoun – second person	তোমার
v.0	Verb – non finite	বলতে
v.3	Verb – third person	বলছে
noun	Noun	পানি
no	Determinator - No	না
ign	Ignore	আজকে

3.2 Suffix Processing

In Bangla, suffix plays an important role in detecting the person of the word as well as suffix tag is also used to modifying a word properly to find out the stem or root. For adding the suffix tags, there is also a dataset. In the dataset, for each same type suffix, there is a corresponding tag. These tags are helpful to detect the input sentence type and to parse the main information. The suffix tag is added with the POS tag. Some

suffix tag is described in Table 2. For simplicity, we treated pronoun as a noun and it can also have a suffix tag.

Table 2. Example of suffix tag

Suffix Tag	Suffix(es)	Example word
ke	কে, রা, টা, টি, য়, গো, তেই	আকাশটা
er	ের, র, টার	করিমের
e	েই, ই	হাসাই
eto	িত	আলোচিত
null	-	দান

3.3 Person Mapping

As we discussed earlier, in Bangla, the suffix of verb changes depending on the grammatical person. As an example, the first person word ‘যাচ্ছি’ will be changed to a third-person word ‘যাচ্ছেন’. Again, the pronoun also changes depending on the speaker and there is a different subcategory for each person. The word ‘যাচ্ছেন’ can be in the form of ‘যাচ্ছো’ or ‘যাচ্ছিস’. So, a mechanism is required to handle this type of complex scenario. We introduced a static mapping mechanism that applies to the most type of words. A dataset is made which stores the static mapping of the suffix or person change. For pronouns, direct word mapping like ‘আমাকে’ to ‘তোমাকে’ is added there. Some mappings are given in Table 3.

Table 3. Example of person mapping

Suffix/Word		Example	
From	To	From	To
িনি	েননি	শুনিনি	শুনেননি
াম	েন	জানতাম	জানতেন
ো	েন	বলবো	বলবেন
-	েন	চল	চলেন
আমাকে	তোমাকে	-	-

3.4 PosNeg Mapping

TUNI can reply more naturally with expressions using some interjections like in Bangla, ‘বাহ!’, ‘সুখবর!’ etc. To make a proper expression the meaning of the input sentence must be evaluated. For our method, we proposed a basic sentiment analyzer that can detect the type of the sentence so that she can make expression accordingly. For detecting a sentence contains whether a positive meaning or negative meaning or

none are done by checking the presence of some words. The words are stored in the PosNeg dataset. So, for each word in the dataset, there is a corresponding tag that suggests if the word has a positive or a negative meaning. A snip of the dataset is shown in Table 4.

Table 4. Example of PosNeg tag

PosNeg Tag	Description	Example words
p	positive	ভালো, বেশি, সুখ, বেশি
n	negative	খারাপ, মৃত্যু, দুঃখ, কম
x	neutral	আমি, তুমি, ঘুম

3.5 Sentence Deviation Processing

Breaking down a complex or a compound sentence is necessary for simplifying the information parsing operation. A sentence can be broken down based on some words or punctuations and its position in the sentence. The words or punctuations are stored in the Sentence-Divider dataset. Some contents of the dataset are: ।, comma, !, তাহলে, তবুও, যে, যখন, কারন etc.

4 Corpus Structure

A corpus is a large and structured set of texts [12]. Our corpus contains mainly four pieces of information. They are keyword, precedence of the keyword, decomposition rules and reassembly rules. A keyword can be a collection of Bangla words or parts-of-speech tags. The words are divided by slash. Each keyword contains a precedence value which is an integer number that determines the importance of the keyword in a sentence. The precedence values are well observed and statically given by us. Each keyword contains a list of decomposition rules. Decomposition rule is made of some criteria which can be a POS tag, or a direct Bangla word or both. Multiple POS tags and direct words are separated by a slash. For each decomposition rule, there are several reassembly rules. Reassembly rule contains some numerical notation which specifies the index of the parsed information.

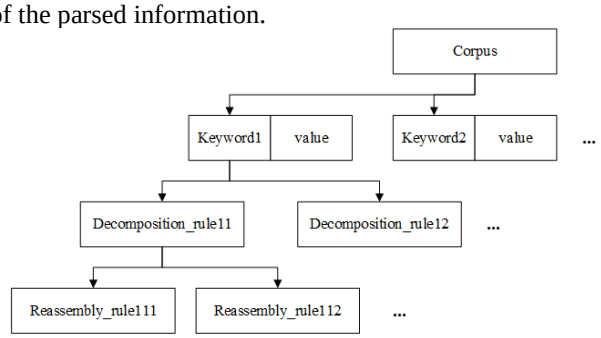

Fig. 1. Simplified corpus structure

5 Methodology

After getting the input text, the text is preprocessed which makes the input text ready for the next step, finding focus. In the finding focus step, the best word which represents the meaning of the sentence is chosen. Then, in the matching step, the preprocessed sentence is checked with all the decomposition rules within the focus. The next step is parsing, where the main information is parsed from the input with the help of the decomposition rule. After that, the reassembly step begins where the parsed information is used to make a relevant reply. The flowchart is shown in Fig. 2.

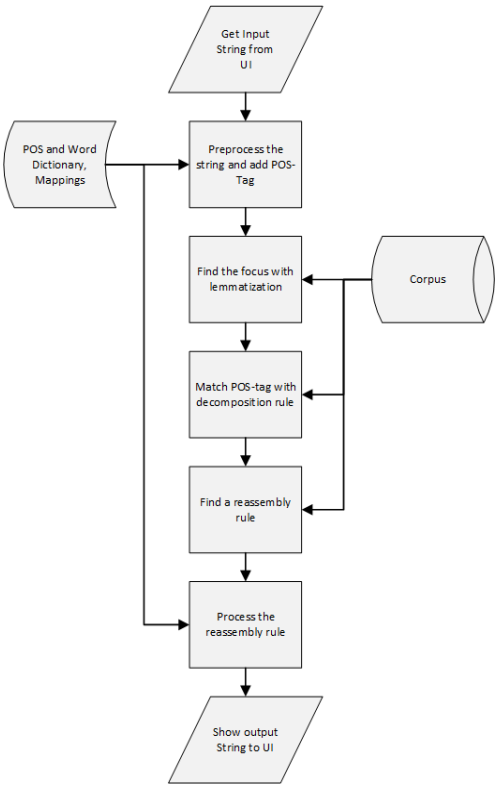

Fig. 2. Flowchart of TUNI

5.1 Preprocessing

This is the first step of the proposed method. Five basic operations are performed in this step which helps to modify the input sentence such that it can be used in the next steps. So, other steps completely depend on this step. For the input sentence "আমি এখন পড়ালেখা করছি কারন আমি ভাল ফলাফল করতে চাই" we can see the step by step pro-

cessing which will help us to illustrate the methodology easily. The five operations are described below:

Removing unnecessary spaces: If the input string contains unnecessary spaces, then all of them are replaced with a single space. If the unnecessary spaces are at the beginning or at the end of the sentence, then they are just removed. Space is added before every punctuation or symbol of the input text.

Tokenization: The tokenizer for our context is need not to be very advanced. In Bangla, the tokenization process is simple because only space is enough to find out the tokens from the input sentence. The tokens are saved into a list named token-list. For the above example sentence, the token-list illustrated in Fig. 3.

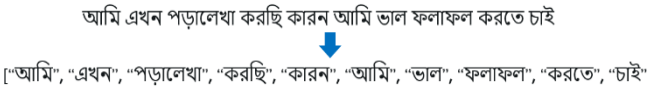

Fig. 3. Tokenization Example

5.2 Breaking Into Sub-sentence

A sentence may be complex or compound but there can be only one focus in our context. So, breaking down this type of sentence is necessary to find out the focus and the main sub sentence for meaning parsing. A complex or compound sentence can be broken down based on the presence of some words or punctuation which are described in the Sentence-Divider dataset. After tokenization, the token-list divided into sublists depending on the sub sentence. For our above example, the input sentence is a compound sentence, so it needs to be broken down. The sentence contains the word "কারন", which is in the Sentence-Divider dataset and the input sentence will be broken down based on it. After breaking down, the token-list is sub divided into two or more lists. For the previous example, it is illustrated in Fig. 4.

Fig. 4. Sentence breaking into sub-sentence

5.3 Adding Tags

For each of the token of the token-list is added with it's corresponding POS tag using the POS tagging dataset which is described in 0After adding the POS tags, suffix tag is added with 'noun', 'verb' and 'pronoun' POS tags. Suffix tags are described in 0 For our example, the addition of POS and suffix tag is illustrated in Fig 5.

[[“আমি:p.1.null”, ”এখন:ign”, “পড়ালেখা:noun.null”, “করছি:v.1”],

[“আমি:p.1.null”, “সিজিপিএ:noun.null”, ”বাড়াতে:v.0”, “চাই:v.1”]]

Fig. 5. POS and suffix tag illustration

5.4 Removing Unnecessary Words

Unnecessary words that don’t play an important role in the meaning of a sentence can be ignored for the next steps. These type words are annotated with ‘ign’ POS tag in the previous POS tagging step. So, tokens with ‘ign’ POS tag, are removed from the token-list. For our example, the token ‘এখন’ is with the ‘ign’ POS tag, and it can be removed from the token-list.

5.5 Finding Focus

Focus is the main token which plays an important role in the meaning of a sentence. In our corpus, there is a precedence value for each keyword. The higher is the value, the more importance in a sentence. For each token in the token-list, it is searched in the corpus for precedence value. The token may not be the same as the keyword in the corpus, so if any token isn’t found in the exact search, then the stem of the token is searched in the corpus. Again, if not found, then the POS tag of the token is searched in the corpus. If no POS tag is found corresponding to the word, then precedence value ‘-1’ is set. The search tree and the value assignment are shown in Fig. 6. When all values are assigned to all tokens, then the token with the highest precedence value is selected as focus, and the sub sentence containing the focus is selected as a new token-list. For our previous example, the token ‘চাই’ has the highest precedence value, 4. The selection is illustrated in Fig. 7.

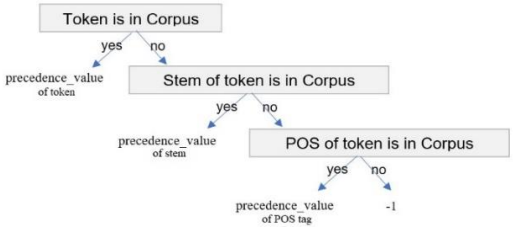

Fig. 6. Keyword search tree

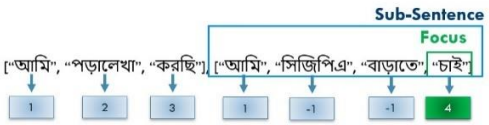

Fig. 7. Focus and sub-sentence selection illustration

5.6 Matching

For each keyword in the corpus, there are some decomposition rules, we can see in Fig. 1. When a focus is selected, all of the decomposition rules are checked with the token-list until one is matched. So, the rule which is matched first is selected for parsing. Decomposition rule is a list of some conditions which must be met with the token-list. Conditions are the combination of POS tags or direct word or PosNeg tag. As an example, if in the decomposition rule, there is a condition 'v.0', then to be matched with this rule, the token-list must have a token with 'v.0' POS tag. The token-list may have more POS tags than the matched decomposition rule which are ignored. It is noted that the decomposition rule doesn't contain any condition regarding the focus. Continuing with the previous example, the matching step is illustrated in Fig. 8. Here, the first rule doesn't match because the token-list has not any word 'হতে'. The rest of the rules do match but, the second rule is selected as it is matched first. The last rule – '-' means the null condition and can be matched with any token-list.

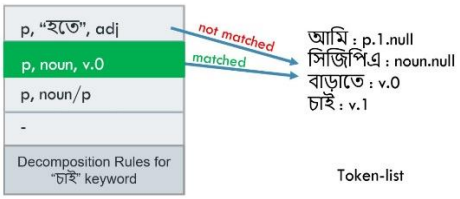

Fig. 8. Matching illustration

In this step, a simple sentiment of the token-list is evaluated if there is any condition of sentiment in decomposition rule. The sentiment tags or PosNeg tags are shown in TABLE IV. Every token of the token-list which is in the PosNeg dataset is selected for processing. Then the token with positive tag is set with value '+1' and the negative token is set with '-1'. After multiplying them, if the result is '+1' then the input string considered as positive meaningful sentence otherwise it is a negative meaningful sentence. When no condition regarding sentiment is in the decomposition rule or no token of the token-list in the PosNeg dataset then the sentence is not evaluated and set as neutral. For our previous example, no PosNeg tag is in the decomposition rule, hence no sentiment evaluation is performed.

5.7 Parsing

When a decomposition rule is selected, it is selected for parsing the token-list. A dictionary is used to store the parsed tokens corresponding to the conditions of the decomposition rule. So, the key of the dictionary is condition and the value is token. If there is more than one token for a condition, a list is used as the value. In this case, the order of the tokens in the list follows the order of the token according to the POS tags in the decomposition rule. For the same previous example, the parsing step is illustrated in Fig. 9.

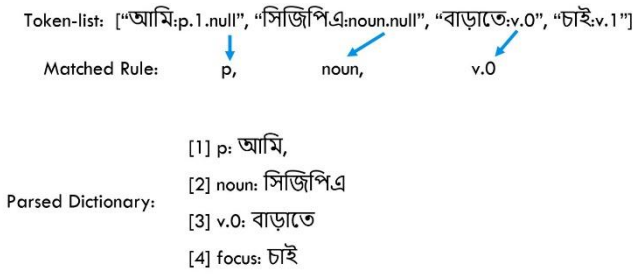

Fig. 9. Parsing illustration

5.8 Reassembly Rule Processing

With the parsed dictionary, a reassembly rule integrates the parsed token to produce meaningful replies within the context. For each decomposition rule, there are several reassembly rules in the corpus shown in Fig. 1. This step is sub divided into three steps:

Rule Selection: Reassembly rule is selected circularly so that TUNI responds more naturally with the same input. A dictionary is maintained which stores the index of reassembly that just has been used. For every response, the index is incremented by the selected decomposition rule.

Placing Token: Reassembly rule contains some numeric notation. They are the index of the token in the parsed dictionary. The notations are replaced by the corresponding tokens. There is a 'foc' keyword which means the matched focus and it is replaced by focus word. For our example, it is illustrated in Fig. 10, where both '1' and '2' are the numeric notation and replaced by 'আমি' and 'সিজিপিএ' respectively from the parsed dictionary.

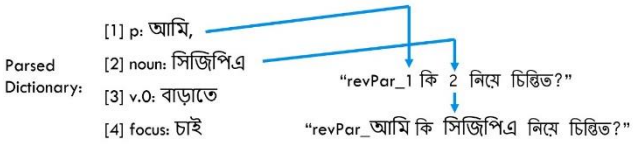

Fig. 10. Placing token illustration

Evaluating Functions: Evaluation functions can modify parsed information as needed. They are described in Table 5. The 'revPer' and 'rev_all' functions use the person mapping dataset described in Table 3. When all of the functions are evaluated then the string is ready for output. For our example, function evaluation is illustrated in Fig. 11. In this example, only 'revPer' function is available for the reassembly rule which modifies the 'আমি' token into 'আপনি'.

Table 5. Reassembly functions

Function	Description
revPer	Reverses the person form of specified token or word
addSuff[]	Adds or replaces suffix as required
rev_all	Reverses all the person form in a sentence
goto	Selecting replies from another similar keyword

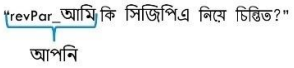

Fig. 11. Function evaluation illustration

6 Experimental Results

Since there are only a very few conversational agents available in Bengali, we will compare our system with English conversational agent, ELIZA. ELIZA is also a corpus based system like TUNI. The comparison is shown in Table 6. From the table, we can see for the first case ELIZA replied with an inaccurate sentence where TUNI replied with an accurate and more natural sentence. For the rest of the cases, we can easily differentiate the responses from ELIZA to TUNI and say that TUNI performs well.

Table 6. Comparison between ELIZA and TUNI

ELIZA	TUNI
YOU: I can't sleep from few days. ELIZA: Perhaps you can sleep from few days	আপনিঃ আমি কিছুদিন থেকে ঘুমাতে পারছি না TUNI: কেন আপনি ঘুমাতে পারছেন না?
YOU: an accident has happened with me. ELIZA: What does that suggest to you?	আপনিঃ আমার সাথে একটি দুর্ঘটনা ঘটেছে TUNI: কি দুর্ঘটনা ঘটেছে আমি কি জানতে পারি?
YOU: my cousin died in front of me. ELIZA: Your cousin died in front of you?	আপনিঃ আমার চাচাতো ভাই আমার সামনে মারা গেছে TUNI: আমি দুঃখিত, আপনার চাচাতো ভাই কি আপনার খুব কাছের ছিল?
YOU: I want to be happy but depression never ends ELIZA: What if you never got to be happy?	আপনিঃ আমি ভাল থাকতে চাই কিন্তু বিষণ্নতা কাটছে না TUNI: কি হবে যদি ভাল থাকতে না পারেন?
YOU: he wants to buy a car ELIZA: I'm not sure I understand you fully.	আপনিঃ সে গাড়ি কিনতে চায় TUNI: কি হবে যদি সে গাড়ি কিনতে না পারে?
YOU: I will kill him. ELIZA: That's quite interesting.	আপনিঃ আমি তাকে হত্যা করব TUNI: আপনি কাকে হত্যা করবেন?

7 Conclusion

In this paper, a new method for Bangla sentence processing and meaning parsing is proposed. With this method, the main complexities in Bangla language like various person forms, various sentence structures, various suffixes etc. are overcome. Corpus and various datasets have been used for the method. Corpus holds some keywords, rules and datasets hold tags or mappings. And, the datasets hold the dynamic mappings and keywords. An interactive user interface is also implemented for easy communication with TUNI. By using it, a user can type and talk with TUNI. The accuracy of TUNI is depending on the POS tag, suffix tag, PosNeg tag and reassembly rules. The more accurate the tags and rules, the more correct and natural replies are given by TUNI.

References

1. S. Diederich, A. Brendel, and L. Kolbe, “On Conversational Agents in Information Systems Research: Analyzing the Past to Guide Future Work”, pp. 1-3, 2019.
2. S. Hussain, O. Sianaki, and N. Ababneh, “A Survey on Conversational Agents/Chatbots Classification and Design Techniques,” pp. 946–956, 2019.
3. J. Weizenbaum, “ELIZA-A computer program for the study of natural language communication between man and machine,” Commun. ACM, 1966.
4. List of languages by number of native speakers, https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers, last accessed 2020/07/29.
5. R. Xiao, “Corpus creation,” in Handbook of Natural Language Processing, Second Edition, pp. 3-9, 2010.
6. Bengali language, https://en.wikipedia.org/wiki/Bengali_language, last accessed 2020/07/29.
7. M. N. Hoque and M. H. Seddiqui, “Bangla Parts-of-Speech tagging using Bangla stemmer and rule based analyzer,” in 2015 18th International Conference on Computer and Information Technology, ICCIT 2015, pp. 440-441, 2016.
8. R. Wallace, “The Elements of AIML Style”, pp. 41-45, 2003.
9. J. H. Gennari et al., “The evolution of Protégé: An environment for knowledge-based systems development,” Int. J. Hum. Comput. Stud., pp. 115-117, 2003.
10. A. Deshpande, A. Shahane, D. Gadre, M. Deshpande, and P. M. Joshi, “a Survey of Various Chatbot Implementation Techniques,” Int. J. Comput. Eng. Appl., 2017.
11. M. Kowsher, F. S. Tithi, M. Ashraful Alam, M. N. Huda, M. Md Moheuddin, and M. G. Rosul, “Doly: Bengali Chatbot for Bengali Education,” in 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019, 2019.
12. Text corpus, https://en.wikipedia.org/wiki/Text_corpus, last accessed 2020/07/29.