ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Website: www.ijircce.com*

Vol. 6, Issue 5, May 2018

# NaturalJava: An Efficient Interface for Interpreting Natural Language into Java Code

Ashwini G. Bari[1], Mohit L. Patil[1], Gayatri V. Gaikwad[1], Sushil D. Karandikar[1].

U.G. Student, Department of Computer Engineering, SSBT College of Engineering & Technology, Jalgaon, Maharashtra, India.[1]

**ABSTRACT:** Programming is used in wide variety of domains like astronomy, industrial automation, financial analysis, microbiology etc. People have good logical skills along with great algorithmic solution designing capabilities but due to lack of knowledge of programming languages make them handicapped. NaturalJava user interface supports for creating and modifying Java programs. The interface exploits three subsystems, preprocessing and POS(Part of Speech) Tagging that extract verb, ad-verb, nouns and discard unwanted data. Another subsystem performs matching of tokens and third one generate AST(Abstract Syntax Tree) from which java code is generated.

**KEYWORDS:** AST(Abstract Syntax Tree), POS(Part of Speech) Tagging, Natural Language Processing, User interface.

## I. INTRODUCTION

Programming and Information Technology has been playing a vital role in the rapid growthand transformation of today. Algorithms form the fundamental blocks of programming andare core to solution designing. Implementation of these algorithms using programming languages serves as a major hurdle. Though people have good logical skills along with greatalgorithmic solution designing capabilities but due to lack of knowledge of programminglanguages make them handicapped. It is difficult for beginner programmers to learn syntactical skills and general programming skills simultaneously. Therefore it is necessary todevelop software that is capable of converting algorithms written in natural language to aprogramming language.

Sentences as input
NaturalJava
Java code as output

Fig. 1.1: NaturalJava

## II. LITERATURE SERVEY

David E. Price in[1] developed a prototype text-based natural language interface for Java programming that accepts English sentences from the keyboard and produces syntactically correct Java source code. The NaturalJava prototype possesses a number of limitations resulting from our depth first development strategy. Its interface is very simple, allowingonly input typed at the keyboard. It is best suited to program creationediting features areextremely limited. NaturalJavas input and output languages are limited in several ways. Because It have limited English vocabulary which is used to describe programs. Additionally,
Sundance has problems correctly parsing some of the unusual grammatical structures thatmight be used in programming. For output, NaturalJava supports most of the core Javalanguage, but supports little of the Java API.

Ellen Rilof in[2] the natural language processing group at the University of Utahhas been developing an in-house natural language processing (NLP) system called Sundance.Sundance was originally developed to provide basic sentence processing and information extraction capabilities for the NLP research efforts at Utah. A comprehensive shallow parser performs partial parsing as well as it performs some additional operations like sentence processing, which is needed for information extraction for achieving. As a result, Sundance does substantiallymore than a typical shallow parser, but it still adheres to the shallow parsing paradigm (i.e.,there is no attempt to generate full parse trees or handle deeply nested structures). Shallow parser is faster and robust, but still it is less intending than full parser.
Shusen Li in[3] describes a NLI which takes an ontology knowledge base and NL queries in Chinese as input and outputs SPARQL queries. The system first transforms NL queries into ontology semantic trees based on the ontology, then generates corresponding SPARQL queries, and outputs direct answers from the knowledge base at last. The ontology semantic trees can help representing the semantic structure of NL queries. Experiment results show that the system can represent semantic structure effectively and perform well on spoken Chinese.
Jakub Dutkiewicz in[4] describes an information extraction methodology which uses shallow parsing. It presents detailed information on the extraction process, data structures used within that process as well as the evaluation of the described method. The extraction is fully automatic. Instead of machine learning it uses predefined frame templates and vocabulary stored within the domain ontology with elements related to frame templates. The architecture of the information extractor is modular and the main extraction module is capable of processing various languages when lexicalization for these languages is provided. There are basically two methods of information extraction. The first method, open extraction systems based on statistical classifiers and machine learning are scalable, but not very accurate. Second is rule based, domain specific systems that use linguistic patterns are more accurate.
Suvam Mukherjee in[5] introduced ALGOSmart is an interpreter which converts pseudo which is written using XML to the programming language source code which is in C and Java. But the ALGOSmart interpreter forces an extra overhead on users by making it compulsory for them to have knowledge about a set of predefined XML tags and their correct implementation and use.
David Price in[6] proposed other system called NaturalJava. It is natural language based user interface which allows a user to enter algorithm in natural English language and it provides the corresponding Java code. The archetypical implementation of the proposal mentioned above, called NaturalJava, Have three main modules. PRISM allowed the user to input an algorithmic statement in English language which was in turn passed to Sundance.Sundance then generated the corresponding case frames which were analyse by PRISM by calling Tree Face. After each operation Tree Face generated the source code which was then presented to the user by PRISM. The major limitation in the aforementioned model is imposed by the finite set of case frames and the ability to comprehend natural language input.
Prof. Swapnali Kurhade in[7] proposal was Semi Natural Language Algorithm to Programming Language Interpreter. This translator converts algorithm in natural English language to code in C and Java This interpreter has many semantic challenges such as it does not support multiple variable declaration, it also does not support printing the value of variables. Such limitations imposes constraint on user while developing fully functional
program.

## III. PROPOSEDSYSTEM

Many approaches can be used to convert algorithm to program. Each approach has its own advantages and disadvantages. We have develop a user interface, which is Natural-Language based. It allows user to create Java Program by making use of parsing along with natural language processing and database consisting of build in data types and matching keywords. Input to this purposed system will be statements in natural language (Pseudo code). Output is the file containing equivalent java program.

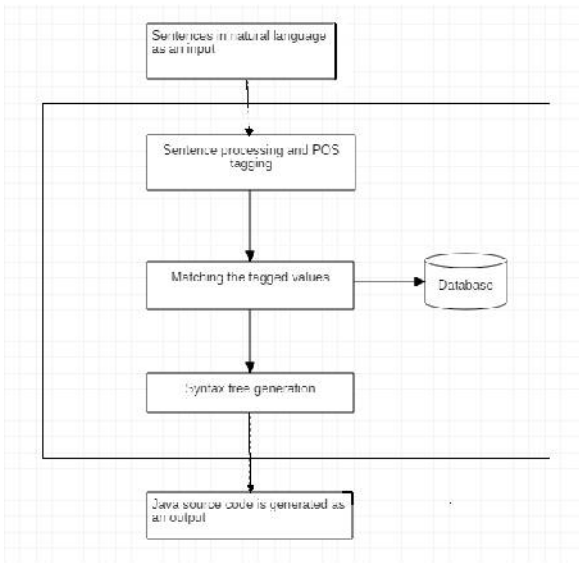

Sentences in natural language as an input
Sentence processing and POS tagging
Matching the tagged values
Database
Syntax tree generation
Java source code is generated as an output

Fig. 2.1: Proposed System

The process of converting the pseudo code (sentences as input) to a java equivalentprogram is by inspecting each and every keyword present in the individual lines.

## IV. NATURALJAVA USER INTERFACE

Grappling with the syntax of a programming language can be frustrating for programmers because it distracts from the abstract task of creating a correct program. Visually disabled programmers have a difficulty with syntax because managing and detecting syntactic errors requiresvisual concentration. ENJA user interface allows programmers to create, modify,and examine Java programs. English sentences are used to describe programs by programmers and the system will automatically builds a Java abstract syntaxtree (AST). When the programmer is finished with his English sentences, Java source code is generate by using AST. The Java program in evolution is also displayed in a separate field duringthe programming process so that it will help the programmer to analyse the code as it is being generated.

### 1. MOTIVATION AND BACKGROUND

Natural language (NL) interfaces can be obsessed with two types of problems,ambiguous and incomplete natural language specifications, and complete NL understanding is still a big problem. The first problem is addressed by limiting the role of inference in system. NaturalJava readsNatural Language commands. Thesecommands are similar to actual programming structure. These are expressed in English. This specification is relatively defined well, so that the programmer can focus on logic instead of syntax. User interface detects when a command is incomplete and prompt the user(e.g., the terminating condition of a loop is missing). The role of user inference is mainly eliminate to the ambiguity of general verbs (e.g., add can implemented as arithmetic or insertion).

The second problem addressedof fragile natural language processing by using information extraction technology supported by POS tagging. POS tagging process is more robust than full parsers. Full parsers often get failed on sentences which are wrong-constructed or grammatically improper. POS tagging is more robust because they do not have to generate a complete AST structure, but instead generate at syntactic representation of sentence fragments. NL interfaces have been developed previously are very few.In a real programming language using an AST,NaturalJava allows users to generate and manipulate source code is the biggest difference between NaturalJava

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

and previous systems. ALGOSmart is one of the older techniques in that ALGOSmart is an interpreter which converts pseudo which is written using XML to the programming language source code which is in C and Java.

## V. IMPLIMENTATION

The operation of NLP to Java convertor system is given below.

1. Start

2. Command/Query processing

At very first, user of this system should have to enter commands one by one for the processing. These commands should be logically and syntactically correct. User willinsert/type one command at a time. Then the inserted command will be processed by clicking 'Process Command' button. After clicking process command button, system does the elicitation of the part of speech tags(POS tags). For this purpose system takes help of standard NLP(Natural Language Processing) libraries. After drawing out this
POS tags, system stores them into the dedicated vectors(for eg. nounAct is a vector used for the NN tag).

3. Tags filtering

All the POS tags that are drawn from the user command in earlier module(command processing) are get filtered in this module. This module uses the database for the filtration process. The database contains two tables. System refers these two tables for the filtration process. These tables are for verbs and nouns in the commands. After drawing out all POS tags, user can filter all these tags by clicking 'Filter Verbs and Nouns' button. After clicking this button, system validates the nouns and verbs from the POS tags. System discards all the POS tags which are not necessary. System keeps only those POS tags which are required for the logic of code generation. These valid POS tags are then displayed on system screen.

4. Syntax tree generation

After filtering the POS tags, user then inserts the data in the node of syntax tree. Usercan insert node by clicking 'Add data to tree' button. The syntax tree is generated as per the logical flow of the logic of the java source code. User can insert child node for any node. It can be done by selecting particular node before clicking add data button. The syntaxes are arranged by using the parent child hierarchy of tree. After clicking 'Add data to tree' button system performs main task. In this task the actual conversation of natural language statement is done. For every particular command system builds actual java syntax. For generation of these syntaxes, system uses particular logic. These logics are chosen according to the combination of POS tags in command. The generated syntax tree is displayed on system screen.

5. Java source code generation

Syntax tree gives the architecture of the java source code that user wants to build.After generating proper syntax tree user can generate java source code. To generate java source code, user must have to process 'terminate class' command. Terminate class command means stop generating further code. Java source code is generated from the syntax tree which is generated by processing all commands. The java code is displayed on the text area field on the system screen. User can make alterations/changes in that code as per his requirements.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Website: www.ijircce.com*

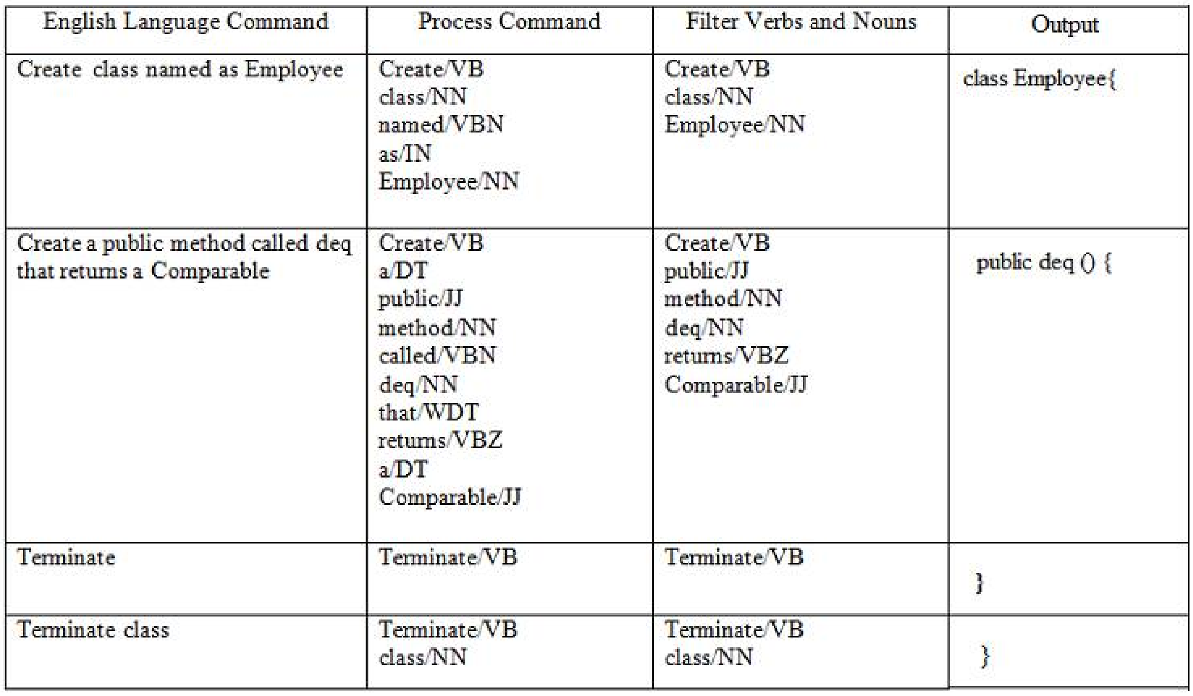
Vol. 6, Issue 5, May 2018

6. Save source code

User can also save this code by clicking 'Save code to file' button. After clicking thisbutton system will prompt save window.

## VI. RESULT

The main purpose of the proposed system is to generate java program code from sen-tences taken as input in natural language. Proposed system demonstrates the use of Natural Language Processing techniques for extraction of sentence into verb, adverb and nouns using preprocessing and POS tagging, then matching these extraction with build in types and then generating AST (Abstract Syntax Tree) and _nally Java code will be generated from sentences given in natural language, which makes easier to generate java code, for those who are syntactically unknown to java.
Figure 6.1 shows the example of command processing in NaturalJava where english language commands are processed to _lter verbs and nouns from commands/sentences.



| English Language Command | Process Command | Filter Verbs and Nouns | Output |
|---|---|---|---|
| Create class named as Employee | Create/VB<br>class/NN<br>named/VBN<br>as/IN<br>Employee/NN | Create/VB<br>class/NN<br>Employee/NN | class Employee{ |
| Create a public method called deq that returns a Comparable | Create/VB<br>a/DT<br>public/JJ<br>method/NN<br>called/VBN<br>deq/NN<br>that/WDT<br>returns/VBZ<br>a/DT<br>Comparable/JJ | Create/VB<br>public/JJ<br>method/NN<br>deq/NN<br>returns/VBZ<br>Comparable/JJ | public deq () { |
| Terminate | Terminate/VB | Terminate/VB | } |
| Terminate class | Terminate/VB<br>class/NN | Terminate/VB<br>class/NN | } |

Fig. 6.1: Result-Command processing in NaturalJava

The syntax tree is generated as per the logical ow of the logic of the java source code.
User can insert child node for any node. it can be done by selecting particular node before clicking add data button. The syntaxes are arrange by using the parent child hierarchy of tree, from the syntax tree generated by proposed system java source code is generated.

I. Advantages

- Less human e_orts required because there is very less need to focus on syntactic knowl-edge about language.
- Easy to develop java program because all programmers need is logical skills.
- It also supports array implementation.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

- Less time consuming as normally programmers su_ers from repetitive stress duringentering and editing syntactically detailed programs from the keyboard.

II. Disadvantages

- It does not support nested classes, interfaces, etc. because proposed system does notbuilt required AST for it.
- Every programming language has its own features. The aspects of various programming language becomes more difficult to incorporate to be identified and interpreted bynatural language processing.
- Proposed system suffers from NLP limitation (such it does not accept variable names-a, an, the, i, etc.).

## VII. CONCLUSION

NaturalJava - NLP to Java Convertor is a system which converts English statements into java code. This system is a prototype for an intelligent, natural-language-based user interface that allows programmers to create, modify, and examine Java programs. If the algorithms mentioned in natural English language effectively get converted to code will help programmers to focus on the logic building and make them free from syntax worries, further it will also aid the visually impaired programmers. NaturalJava overcomes the problem of array declaration and nested class which are with the existing system.

## REFERENCES

1. David E. Price, Ellen Rilof, Joseph L. Zachary,“A Study to Evaluate a Natural Language Interface for Computer Science Education”, National Conference on Atrifitial Intelligence, ”Inquiry Based Learning Science”, pp 49-58, 2007.
2. Ellen Rilof and William Phillips,“An Introduction to the Sundance and AutoSlog Systems”, University of Utah, Technical Report UUCS-04-015, 2004.
3. Shusen Li, ZhiyangHe, Ji Wu, “An Ontology Semantic Tree based Natural Language Interface”, IEEE , 978-1-4799-4219-0, pp 226-230, 2014.
4. Jakub Dutkiewicz, Maciej Falkowski, Maciej Nowak, and Czesaw Jdrzejek, “Semantic Extraction with Use of Frames”,Springer International Publishing Switzerland, pp 208-215, 2014
5. Suvam Mukherjee and Tamal Chakrabarti, “Automatic Algorithm Specification to Source Code Translation”, Indian Journal of Computer Science and Engineering, 2011 on, Vol. 2, No. 2, pp. 146 159, April May 2011.
6. David Price, Ellen Rilof, Joseph Zachary and Brandon Harvey, “NaturalJava:A Natural Language Interface for Programming in Java”, in the Proceedings of the 2000 ACM on Intelligent User Interfaces Conference, pp. 207 211, January 2000.
7. Prof. Swapnali Kurhade and Sharvari Nadkarni, “Semi Natural Language Algorithm to Programming Language Interpreter”, International Conference on Advances in Human Machine Interaction (HMI - 2016), March 03-05, 2016