ALGORITHM TO CODE CONVERSION
USING MACHINE LEARNING

KRISHNA SAI K
DEPT OF INFORMATION TECHNOLOGY
SRI SAIRAM ENGINEEERING COLLEGE
CHENNAI,INDIA
sec20it064@sairamtap.edu.in

DR SURESH KUMAR M
ASSOCIATE PROFESSOR
DEPT OF INFORMATION TECHNOLOGY
SRI SAIRAM ENGINEERING COLLEGE
CHENNAI,INDIA

VARUN ST
DEPT OF INFORMATION TECHNOLOGY
SRI SAIRAM ENGINEERING COLLEGE
CHENNAI,INDIA
sec20it128@sairamtap.edu.in

***Abstract*—The system that has been proposed "Alcode" is essentially a translator or converter that beginners can use to convert a given input algorithm to the appropriate code. Unlike other converters, Alcode has no trouble adapting to different users' writing styles and can do so quickly. Synonyms, a naive base classifier, and an XML Specification File are all used in the system.**

Keywords: Naïve Bayes classifier, XML specification file, mapper phase, trigger,

Introduction

The proposed system Algorithm to Code Converter (Alcode) is an interpreter or a translation process that allows the user to write only the algorithm of a problem in semi-natural English language which can further be translated into either Java or C programming language code. This system will basically help the neophyte users, visually impaired people (who deal with issues of punctuation marks and syntax) and those who find it difficult to write good and efficient programs. An algorithm gives the procedure for solving a problem. Generally, we deduce an algorithmic logic step by step to build a solution and then convert it to code to solve a problem. Sometimes a person may know the steps to solve a program but is not comfortable writing it in a particular programming language like C where syntax is very important.

MOTIVATION:

Inexperienced persons have a hard time remembering a language's syntax and occasionally don't grasp how to write code in a programming language. As a result, implementing an algorithm in such languages is challenging. Even if the logic is right, it is difficult for someone who is not trained in computer science and does not have good coding skills to construct a syntactically accurate programme and produce the intended outcome.

RELEVANCE OF THE PROJECT:

A. TO CONVERT AN ALGORITHM TO A CODE TAKE AN ALGORITHM AS INPUT AND TRAIN IT, TAG IT WITH POS (PARTS OF SPEECH), THEN USE PARSER PROGRAMMES TO GENERATE THE NECESSARY C CODE. ANOTHER SYSTEM NOW IN USE TRANSLATES USER-WRITTEN PSEUDOCODE IN XML FORMAT TO C AND JAVA LANGUAGE CODE. IN THE PROPOSED SYSTEM, PLAIN TEXT WOULD BE INTERPRETED USING THE NAVE BAYES CLASSIFIER, AND INFORMATION WOULD BE EXTRACTED USING TAGS GENERATED USING POS TAGGING. AFTER THAT, AN XML FILE WILL BE CREATED THAT CONTAINS ALL OF THE INFORMATION NEEDED TO PRODUCE THE ASSOCIATED CODE.

B. ***OUTLINE OF THE REPORT****:*

C. *From the input algorithm, firstly an XML Specification file will be generated. This file will contain all the information required to generate correct code. The user need not to bother about any XML implementation details here because internally this file will be created with no requirement of the user's interference in this phase. This generated intermediate XML Specification file can be converted into any programming language using a mapper program. Different people have different styles and ways of writing algorithms.*

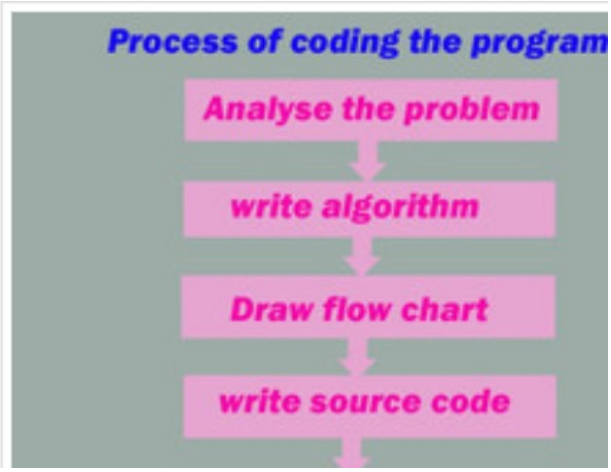
EXISTING SYSTEM :

In the present process of coding the programs, we need to write the algorithm, flow chart, source code for the program. It is very time taking process and we will face many difficulties while coding them. The drawbacks of this kind of process are:-> Time taking

-> Facing many difficulties while coding

-> Un easy to understand and code by a normal person.

Therefore in order to cover the above set of problems, this converter is introduced.



ALGOSmart[1]:

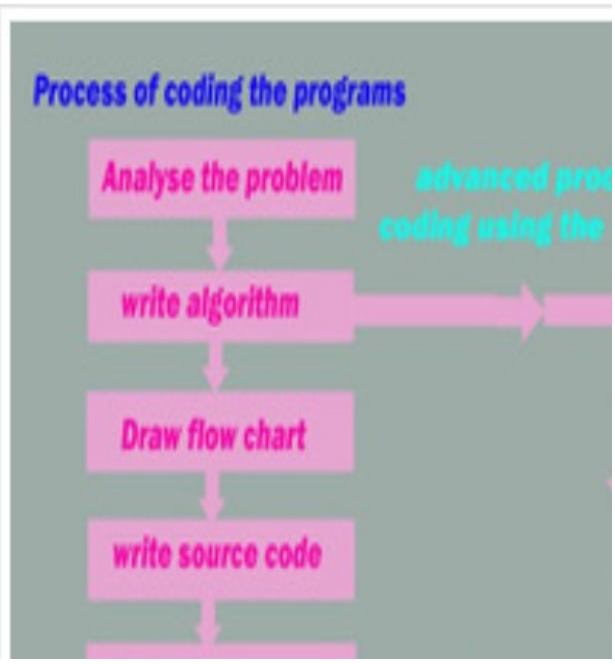
ALGOSmart is a converter that accepts the input pseudocode which should be strictly written in the XML format only and then converts it into C and Java language code [1]. But this translator imposes an additional overhead on users by making it mandatory for them to have a prior knowledge of how to write a pseudocode in XML format [1]. This was the major disadvantage of ALGOSmart as it imposed a constraint on the users to learn the XML language first in order to build an algorithm in XML format.

NaturalJava[2]:

It allows the user to take input algorithms that are written in natural English language and it generates the corresponding Java code for that algorithm. NaturalJava, contained three main modules - PRISM, Sundance Parser and TreeFace. PRISM allowed the user to write the algorithm which was passed to Sundance parser. Sundance then generated the corresponding case frames which were analyzed by PRISM by calling TreeFace [2]. After each operation TreeFace generated the source code which was then presented to the user by PRISM [2]. It allows the user to input algorithms in natural English language and it generates the corresponding Java code. The major limitation of this system is that the Sundance parser generated very few finite set of case frames and hence the input statements were not correctly classified [2]. Due to this, the required output was not generated.

PROPOSED SYSTEM:

Let us have a brief overview of our current application. which allows the user to directly convert the algorithm to source code(c code) Due to the drawbacks of the general coding process we use the advanced coding technique using this converter.



The proposed system Algocode converter uses all the best portions of the existing projects to adapt different writing styles of the users and generate an efficient and desired output

The major components include:

1. Algorithm
2. Translator
3. XML Specification File
4. Mapper
5. Desired Code

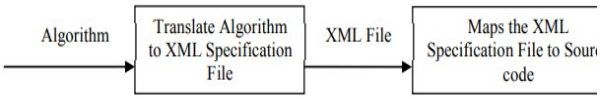
1. Algorithm: This module signifies the input provided by the end user. Via a web-based application, an algorithm is accepted into the systemform of a text file. The users will have to browse the text file from the GUI and upload it.

Step 2 : The entire process of translating an algorithm to any programming language will be divided into two phases.

Identify applicable funding agency here. If none, delete this text box.

Step 3: The class having the highest probability will be selected and the statement categorized as that type.

Step 4: Then, POS Tagging will be performed and the relevant information will be extracted based on the type of statement.



Phase 1-Translator Phase:

Steps involved in the Translator Phase:

Step 1: The algorithm will be read line by line and passed to the Naïve Bayes Classifier.

Step 2: Naïve Bayes classifier will accept the statement and determine the statements' type i.e. it will determine if the statement is conditional, declaration, input or output statement by calculating the probability as to how much does the statement belong to each class of statement i.e. initialization, looping or conditional.

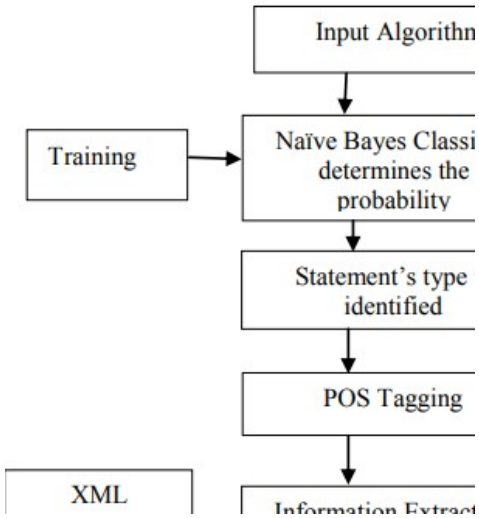
Step 3: The class having the highest probability will be selected and the statement categorized as that type.

Step 4: Then, POS Tagging will be performed and the relevant information will be extracted based on the type of statement.

Step 5: With the help of the information extracted, XML tags will be generated.

Step 6: Performing the above steps on the input statements will generate the XML Specification File.

Step 7: Training will be done to improve the accuracy of Naïve Bayes Algorithm in identifying the statement type.



. ***Converting algorithm to XMl specification file***

Phase 2-Mapper phase:

The mapping process is all about using conditional statements. The parsing of the XML specification file will be carried out by performing simple conditional if-else statements on the tags. We have chosen Python to write this module because Python is particularly well equipped to handle these types of data.

Output of the mapper phase: All the XML tags present in the XML Specification file are mapped to the corresponding C Language code by the mapper. A corresponding C code will be generated and displayed on the screen

Example Input:

Algorithm to find if a number is even or odd

declare integer n
display "Enter a number"
read n from the user
check if n is divisible by 2
display "Number is even"
else display "Number is odd" endif

Generated XML Specification File:

```
<start>
 <declare>
 <datatype>int</datatype>
 <var>n</var>
```

```
<val>0</val>
</declare>
<write>
<value>
        Enter a number</value>
</write>
<input>
<var>n</var>
</input>
<if>
<condition>
            n%2==0 </condition>
<write>
<value>
          Number is even</value>
</write>
</if>
<else>
<write>
<value>
            Number is odd</value>
</write>
</else>
</start>
```

OUTPUT AS C CODE:

```
#include<stdio.h>
void main()
{
 int n=0;
 printf("Enter a number");
 scanf("%d",&n);
 if(n%2==0)
 {
 print("Number is even");
 }
else  {
 print("Number is odd");
 }
}
```

REQUIREMENT SPECIFICATION:

SOFTWARE REQUIREMENTS:

- Visual code studio
- 64 bit windows operating system or Mac OS
- Minimum 4 GB or Maximum 8 GB RAM
- Notepad for algorithm

RESULT AND CONCLUSION:

This system will help the beginners and visuallympaired programmers to write efficient code without worrying about syntax. A2C converter accepts algorithm as input in a normal text file. The entire algorithm will be read line by line till the end. Each line will represent a single operation for the computer to perform. Further, each line will be passed to the Naïve Bayes Classifier program which will interpret the type of statement. In this way, Naïve Bayes will help us to identify if the statement is an initialization, looping or conditional statement. Once the type of statement is identified, POS Tagging will be performed on this sentence through which the required information will be extracted and the corresponding XML Tags will be generated. In this way, an XML Specification file will be produced. Once the XML specification file is generated, it will be mapped to any coding language.

FUTURE SCOPE:

Thus our system can be used by beginners and visually impaired programmers which will help them to focus on logic building and provide them with the desired efficient code for which they have written the algorithm.

And it also helps in a smarter way of programming in a simple words not much more stuffs being provided it improves your thinking skills and able to understand the logic

REFERENCES:

http://codershunt.weebly.com/projects/algorithm-to-code-converter

https://www.semanticscholar.org/paper/Algorithm-to-Code-Converter-Dubey-Edward/204d2de6da020744ae5550cf8d6dc7ac79da1f74

[1]