

**SCHOOL OF COMPUTER SCIENCE**

UNIVERSITY OF PETROLEUM & ENERGY STUDIES

Bidholi Campus, Energy Acres, Dehradun – 248007.

**SYNOPSIS REPORT**

**Text Analysis Using Hierarchical Approach**

**Submitted by**

Yash Jangid (Enroll No. R970218068)

Rishabh Mishra (Enroll. No. R970218044)

Abhinav Kamboj(Enroll. No. R970218067)

Arjun Gupta(Enroll No. R970218010)

BRANCH :- Oil & Gas Informatics

SEMESTER :- V

**Under the guidance of**

Mr. Rahul Kumar Singh

Assistant Professor

Department of Informatics

**June Dec-2020**

**PROJECT TITLE:**

Text Analysis Using Hierarchical Approach .

**OBJECTIVES-**

* + - Preprocessing and tokenization of the given set of words.
    - Part of speech tagging for future extraction.
    - Hierarchical representation of a text for analyses.

**INTRODUCTION**

In this project we have to analyses a given set of texts using various Tokenization approaches in which we can use parsing tree algorithm and other hierarchical approaches for breaking down text document apart into those pieces. We have to take a set a words in any language and analyses them according to the grammar rule and the translate the given set of words in another preferred language in which we want to translate them, by making there parsing or hierarchical tree structures as per there grammar rule in the accordance of verbs, noun, pronoun, etc.

After preprocessing analyzing the set of words accordance to their rules we have to interchange the words in other language, by also maintain there rules , By this approach we can convert the whole text into another language.

Nowadays, The most utilized language in the World is "English". So it's been a provoking undertaking to change over the any language into most helpful language like English. But Machine translation which is known for translating text from one language to another through computers , so, we can easily and efficiently convert the any text into English text. We can also do the same with Human translations but it is not a core solution to this problem. This system is very helpful in teaching and learning any language to understand its features like, unambiguous, well- structured grammar etc. And also helpful for the research purposes.

Whereas Machine translation system is the most beneficial way for this purpose because it includes high translation speed, cheap / less cost, easy to translate into multiple language at once in multilingual environment, store more data in comparison of human brain and most useful benefit is its availability , we can use this system at anytime anywhere.

After building a parser tree in the first phase which is developed through the **Top-Down** approach parser, after that we have to perform a language conversion in which we have to processed the given grammar or a set of words and translate into another language in which we have to translate.

And this translation is also consists or performed by parser algorithm itself, the output which we get from first phase in the form of a tree, we can use it as a input for out second phase and we have to processed the words in a **Bottom-Up** approach of parser, accordance to the grammar rules of context free grammar and of the language in which we have to translate the given language.

**LITERATURE REVIEW**

Our work can be divided into four parts, “Preprocess”, “Tokenization”, “Hierarchical Representation” and “Analyzing accordance to language rules”. In tokenization and preprocess we normally divide the set of words into various segments, i.e. A line consists of 10 words then, we can tokenize the line into 10 tokens and process them in accordance their rules of noun, pronoun, verbs etc. Now, After Preprocessing and Tokenization we can build a hierarchical structure for the same and analyses them according to the rules of another language in which we want to processed the set of words.

After preprocessing, we will get the set of words in an organized manner with respect to grammar rules, them after we have to do a post processing in which we can translate the given set of words according to the language in which we have to convert the given sets, after translation we have to again generate a parser for translated language but this time we can analyses the given parser in bottom up manner, then we have to look at the grammar phase at the time we arranging the words of translated language we according to there grammar, here we can also see that the sentence is not in proper syntax, then we have to make the sets into a proper syntax then we have to combined all the words and our work is done.

Basically, here we are performing analysis that how can we translate a small word set into another language, i.e. here we are performing English to Hindi translation in a generalized manner, after analyzing we perform a basic grammar CFG rules on that and transform it into a proper syntax of sentence.

In the research during the preparation of the project, we found that there were several language divergence approaches to build the same MTS (machine translation system) And to build the same mechanism they were using many technologies like, Corpus-based machine translation, Bilingual evaluation understudy, Canonical syntactic realization, Direct machine translation, Hybrid-based machine translations and much more. For building the same machine translating system they were using many approaches for which a builder needs much knowledge and also cost much. But the same can also be done by simple approach which is lesser in cost also as compare to this high tech. translating systems.

But every project have there Advantages and Disadvantages in high tech. machine translation results in less time complexity and much cost , but as defined above this can be done simply with lesser cost and knowledge but it consumes much time in compression of high tech. MTS.

**METHODOLOGY**

We have to build a simple machine translation system by taking a input as a set of words which can be tokenized and processed further for hierarchical approaches to maintain there grammar rules than again the by applying all these approach we can build another language which also have same meaning as the previous one.

In order to complete the project, we have to divide the project into different segments . This project is divided into six segments.

Gives the introduction about language and Machine translation system.

Defines the various divergence of languages occurs during translation of language with recommendation to handle them.

Then we have to define the various six modules used to proposed Machine translation system.

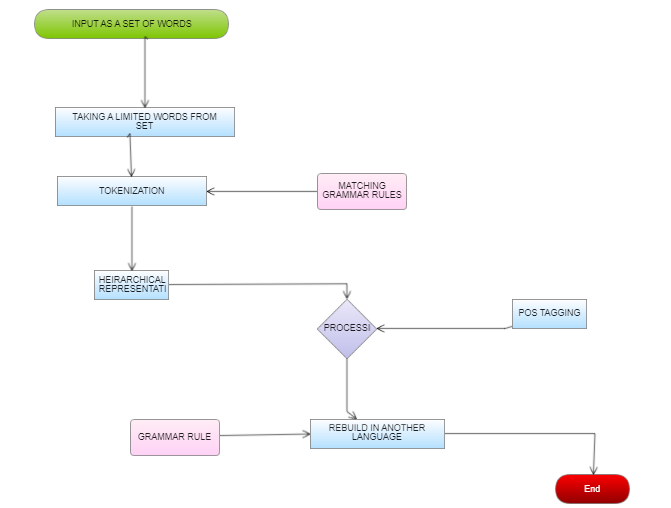
* + - Source language preprocessing
    - POS (part of speech) tagger
    - Parsing table
    - Language tree generation
    - Parse tree generation for parsing table

Fourth segment gives the details of data dictionary, various rules base, tagged corpus and technology used for implementing the purposed system.

Defines the evaluation methods used for the evaluation of the purposed system with the comparison of existing system.

Final conclusion followed by the References (Which is shown at the end of the Synopsis)

**SAMPLE FLOWCHART**



**TEST CASE 1 :-**

In test case we can take a word set by our and and implement it according to the rules which were shown in the Methodology part. We can perform it in several steps which are shown below:-

WORD SET :-

**University of Petroleum and Energy Studies, Dehradun is a renowned University in Northern Part of the India, in the state of Uttarakhand. It is a UGC recognized state private university Established by an Act of the State Legislature of Government of Uttarakhand in the year 2003 and it is listed under Section 2(f) of UGC Act. It is the first Indian University which has got QS 3 star rating overall, QS 5 star rating on the parameters of employability and facilities and a 4 star rating in teaching, in the year 2016.**

**Faculty of any institution is of paramount importance for meaningful impact on the present and future growth of the students as well as on the country. UPES has over five hundred national and international faculty members from highly renowned and leading academic institutes. According to National Institutional Ranking Framework (NIRF), scholarly publication of university faculty members is treated as one of the parameters of University ranking. The present study is the scientometric analysis of scholarly research output of the faculty members of the university.**

The word set define above is a sample set which consists of 176 words or we can say that it contains 176 Tokens , By taking above sample set we can perform Tokenization which is also performed in parts. After tokenization and processing we can build a tree structure or hierarchical structure for the same by maintaining grammar rules and further processed the same tree into another language in which we want to convert.

We can perform the task in some simple steps that are :-

**Step 1 :-** First we have to take a limited word as a Input.

Let,

INPUT :- “University of Petroleum and Energy Studies, Dehradun is a renowned University in Northern Part of the India, in the state of Uttarakhand.”

**Step 2 :-** After taking the Input we have to perform a Tokenization, Input consists of 22 words So we have to divide it into 22 tokens which are:-

“University”, “of”, “Petroleum”, “and”, “Energy”, “Studies”, “Dehradun”, “is”, “a”, “renowned”, “University”, “in”, “Northern”, “part”, “of”, “India”, “in”, “the”, “state”, “of”, “Uttrakhand”.

**Step 3:-** After tokenizing the Input text we have to represent the text in a Hierarchical form or Parse tree representation and for maintaining grammar rule we have to perform the same according to the rules of context free grammar (CFG).

This is the main step among all, So it can be done carefully otherwise it can change the whole meaning of the text.

Before making the parse tree we can define the grammar part accordance to context free grammar

Where,

**NOM =>** University / Petroleum / Energy Studies / Dehradun / Northern part / India / State / Uttrakhand

**Det =>** a / the

**Prp =>** of / in

**Conj =>** and

**Pr** => is

**Adj =>** renowned

S => Par

NOM => 'university' University

Prp => 'of' of

NOM => 'Petroleum' Petroleum

Conj => ‘and’ and

NOM => 'Energy Studies' Energy Studies

NOM => 'Dehradun' Dehradun

PR => 'is' is

Det => 'a' a

Adj => 'renowned' renowned

NOM => 'university' University

Prp => 'in' in

NOM => 'Northern\_Part' Northern\_Part

Prp => 'of' of

Det => 'the' the

NOM => 'India' India

Prp => 'in' in

Det => 'the' the

NOM => 'state' state

Prp => 'of' of

NOM => 'Uttarakhand' Uttarakhand

AS PER THE MODIFIED CFG

S => NP VP

S => AUX NP VP

S => VP

WHERE,

NP => DET NOM

NP => PROPN

NP => PRONOUN

And,

NOM => ADJ NOM

NOM => N

NOM => N NOM

VP => V

VP => V PP

PP => PREP NP

* We can give extension also to make our parse tree simpler :-

S:par

CJT: g(np)

H: n ('university' S NOM) University

D: g(pp)

H: prp('of') of

D: par

CJT: n('Petroleum' S NOM) Petroleum

CO: conj('and') and

CJT: n('Energy\_Studies' S NOM) Energy\_Studies

CJT: n('Dehradun' S NOM) Dehradun

P: v('be' PR S) is

Cs: g(np)

D: art('a' S) a

D: adj('renowned' POS) renowned

H: n('university' S NOM) University

A: g(pp)

H: prp('in') in

D: g (np)

H: n ('Northern\_Part' S NOM) Northern\_Part

D: g (pp)

H: prp ('of') of

D: g (np)

D: ar t('the' S/P) the

H: n ('India' S NOM) India

A: g(pp)

H: prp('in') in

D: g (np)

D: art ('the' S/P) the

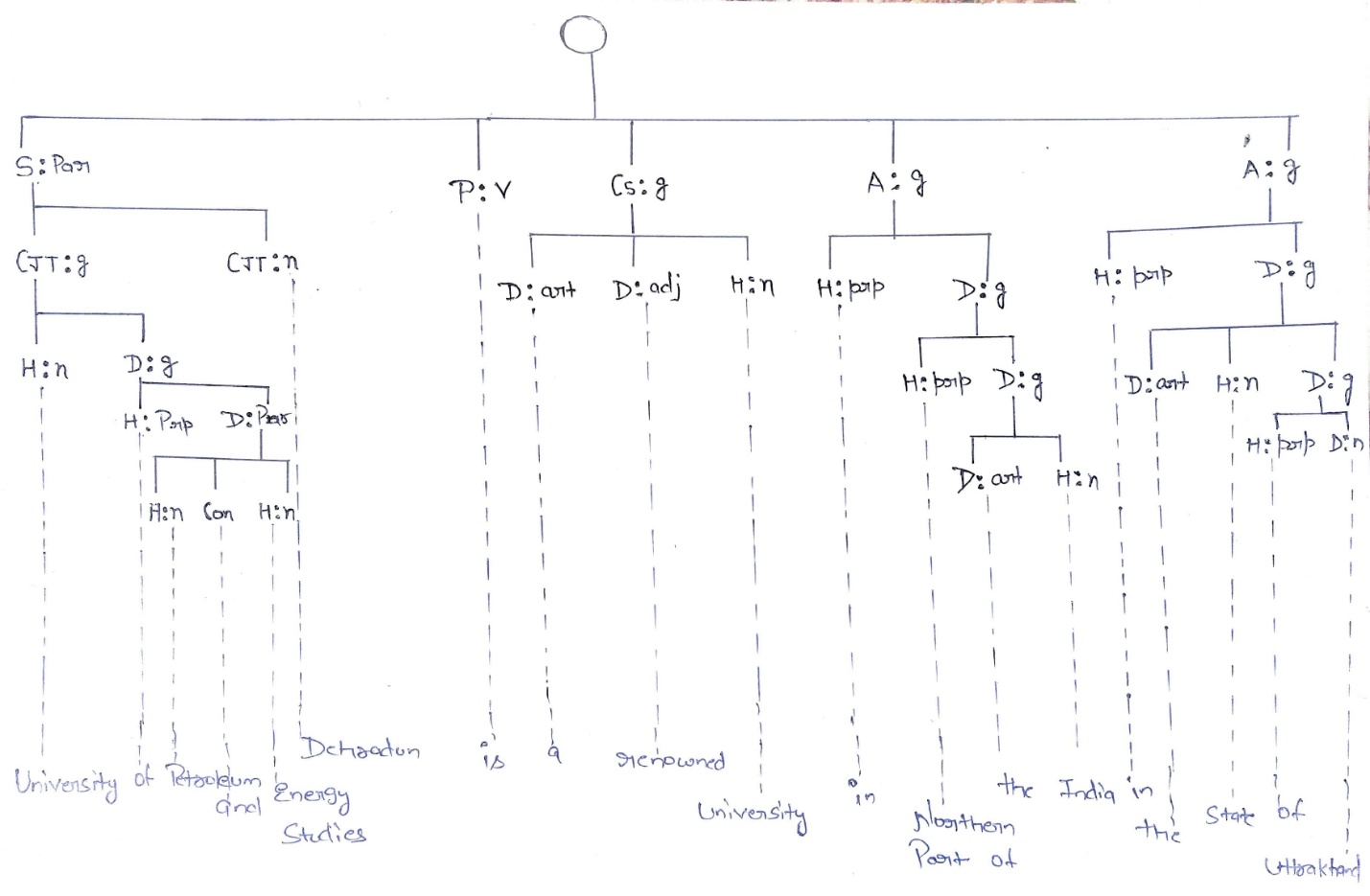
H: n ('state' S NOM) state

D: g (pp)

H: prp ('of') of

D: n ('Uttarakhand' S NOM) Uttarakhand

As per the rules defined , Now we have to design a parse tree for the same:-



**Test case 2:-**

In test case we can take a word set which is produced after the processing of above test case in this final phase we have to transform the set of words in the another language by parsing technique but in Bottom up fashion, after tokenization processed in parse tree and now we can translate those words and here we are transforming the above set in Hindi as per the grammar which is mandatory :-

**Step 1:-**

WORD SET :-

**University of Petroleum and Energy Studies, Dehradun is a renowned University in Northern Part of the India, in the state of Uttarakhand.**

Here, is the word set which same as which we take in the above test case, basically in this test case we can complete the above scenario.

**Step 2:-**

We have to transform the words into another language, for this project we are bound to perform the task under the c language programming guidelines, so here we can perform this phase manually otherwise it can be done easily through machines by using NLP libraries and machine learning.

Here, we can transform the above set in Hindi, so first off all we have to processed the above parser in Hindi then we build a same parser in Hindi to check whether it is according to our grammar rules or not.

**English word set:-** University of Petroleum and Energy Studies, Dehradun is a renowned University in Northern Part of the India, in the state of Uttarakhand.

**Transformed set :-** यूनिवर्सिटी ऑफ पेट्रोलियम एंड एनर्जी स्टडीज देहरादून, उत्तराखंड राज्य में भारत के उत्तरी भाग में एक प्रसिद्ध विश्वविद्यालय है|

Before making the parse tree we can define the grammar part accordance to context free grammar

Where,

**NOM =>** University / Petroleum / Energy Studies / Dehradun / Northern part / India / State / Uttrakhand

**Det =>** a / the

**Prp =>** of / in

**Conj =>** and

**Pr** => is

**Adj =>** renowned

S => Par

NOM => 'university' यूनिवर्सिटी

Prp => 'of' ऑफ

NOM => 'Petroleum' पेट्रोलियम

Conj => ‘and’ एंड

NOM => 'Energy Studies' एनर्जी स्टडीज

NOM => 'Dehradun' देहरादून

PR => 'is' एक

Adj => 'renowned' प्रसिद्ध

NOM => 'university' विश्वविद्यालय

Prp => 'in' में

NOM => 'Northern\_Part' उत्तरी भाग

Prp => 'of' के

Det => 'the' है

NOM => 'India' भारत

Prp => 'in' में

Det => 'the'

NOM => 'state' राज्य

Prp => 'of'

NOM => 'Uttarakhand' उत्तराखंड

Here we can see that the above sentence or action is not correct according to the grammar, so its our main focus to convert the above grammar in a proper sentence in which we can use grammar rules of CFG and in back we also look at the grammar rules of the language in which we have to convert our text.

AS PER THE MODIFIED CFG

S => NP VP

S => AUX NP VP

S => VP

WHERE,

NP => DET NOM

NP => PROPN

NP => PRONOUN

And,

NOM => ADJ NOM

NOM => N

NOM => N NOM

VP => V

VP => V PP

PP => PREP NP

* We can give extension also to make our parse tree simpler :-

S:par

CJT: g(np)

H: n ('university' S NOM) यूनिवर्सिटी

D: g(pp)

H: prp('of') ऑफ

D: par

CJT: n('Petroleum' S NOM) पेट्रोलियम

CO: conj('and') एंड

CJT: n('Energy\_Studies' S NOM) एनर्जी स्टडीज

CJT: n('Dehradun' S NOM) देहरादून

P: v('be' PR S)

Cs: g(np)

D: art('a' S)

D: adj('renowned' POS) एक प्रसिद्ध

H: n('university' S NOM) विश्वविद्यालय

A: g(pp)

H: prp('in') में

D: g (np)

H: n ('Northern\_Part' S NOM) उत्तरी भाग

D: g (pp)

H: prp ('of') के

D: g (np)

D: ar t('the' S/P)

H: n ('India' S NOM) भारत

A: g(pp)

H: prp('in') में

D: g (np)

D: art ('the' S/P) है

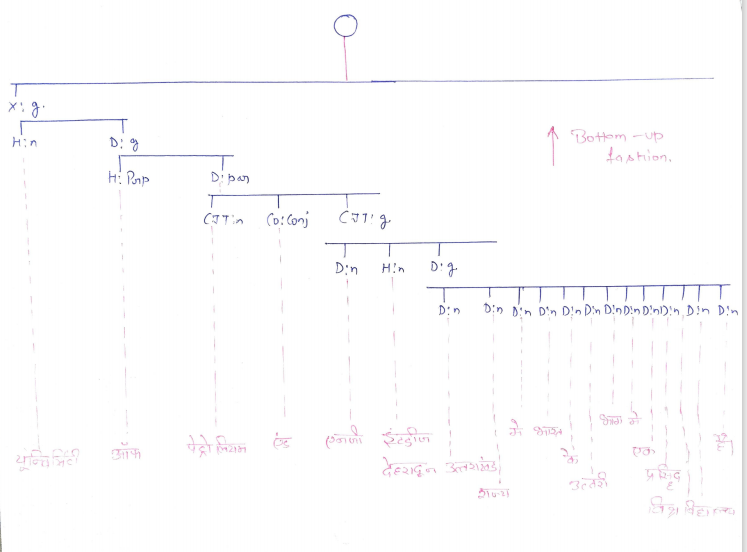
H: n ('state' S NOM) राज्य

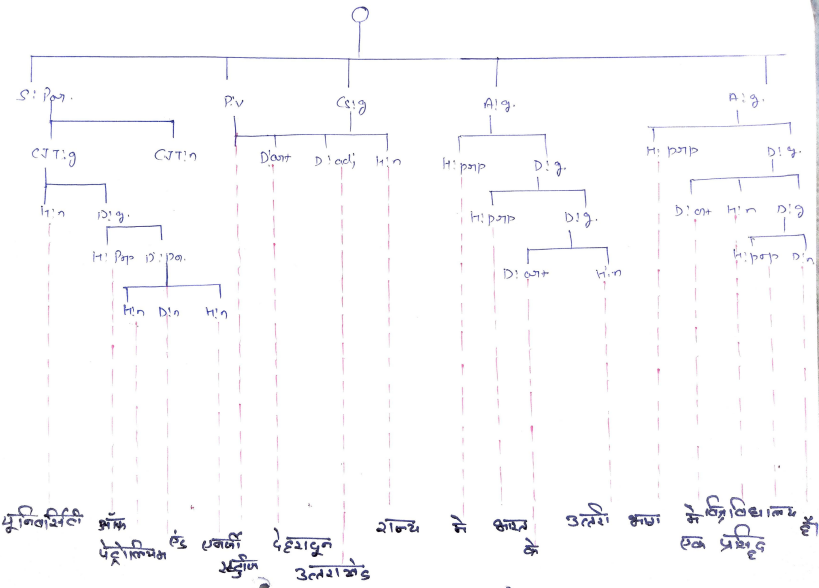
D: g (pp)

H: prp ('of')

D: n ('Uttarakhand' S NOM) उत्तराखंड

As per the rules defined , Now we have to design a parse tree for the same:-

****

****

**NOTE:-**  Finally now we have transferred the given set of words into the another language by performing several tasks.

**ALGORITHM**

1. Start
2. Get the input as a matrix M of n\*n. (n is the number of words in a line)
3. Then, for ( i 🡨 0 to n-1) (i is the number of rows in a matrix M)
4. Print the principal diagonal matrix M as a leaf node
5. Leaf[i] 🡨 M(i,i)
6. Then increment the value of i ( i 🡨 i+1)
7. Take 3 arrays let there name be A, B and C of a size of n-1 sorting of right, left and parent child of the tree,
8. Let, a temporary variable temp = true (initialized)
9. And, m🡨0 (initialize)
10. For (i 🡨0 to n-2)
11. L[m] 🡨 leaf[i]
12. For (j 🡨 i+1 to n-1) where, j is the number of columns in Matrix M
13. If ( M ( i,j) is not equal to NULL and temp = true
14. Then, M(i,j) is the parent node of L[m] i.e. p[m] 🡨 M (i,j)
15. If P[m]=’s’
16. Then, root node = P[m]
17. If j = i+1
18. Then, Leaf [j] is the right node of a tree
19. R[m] 🡸 Leaf [j], and increment the value of m by 1
20. Else
21. M(i+1, j ) is the right node of the tree, and then also increment the value of m by 1
22. i.e. R[m] 🡨 M(i+1,j)
23. And, also update the value of temp as false
24. Else,
25. if ( M (i,j) is not equal to null and the temp value is false )
26. Then, L[m] 🡨 P [m-1] and P[m] 🡨 M (i,j)
27. If, P[m] = ‘s’
28. Then, root node is equal to P[m]
29. i.e. R[m] 🡨 Leaf [j]
30. And, increment the value of m, i, j by 1 also update the value of temp as true
31. For (j🡨9 to n-2)
32. Return (L,P,R) same again.

**Note:-**  Algorithm is modified accordance to the second phase of regenerating, and in both the phase we use same algorithm but in different manner in first phase we used this algorithm in Top-Down manner and in second phase we use same algorithm in Bottom-Up fashion.

**Time complexity**

Best case:

O (n\*n)

* O( n2)

Here, (i , j) > 0

And, i = no. of rows in difference matrix,

j = no. of columns in difference matrix.

**CODE:-**

**For Tokenization of word set:-**

#include <stdio.h>

#include <string.h>

int main()

{

char str1[100];

char newString[10][10];

int i,j,temp;

printf("\nUniversity of petroleum and energy studies.:\n");

printf("\n");

printf(" Input a string for which you want to perform Tockenization : ");

fgets(str1, sizeof str1, stdin);

j=0; temp=0;

for(i=0;i<=(strlen(str1));i++)

if(str1[i]==' '||str1[i]=='\0')

//if space or NULL found, assign NULL into newString[temp]

{

newString[temp][j]='\0';

temp++; //for next word

j=0; //for next word, init index to 0

}

else

{

newString[temp][j]=str1[i];

j++;

}

printf ("\n tokens after split by space are :\n");

for(i=0;i < temp;i++)

printf(" %s\n",newString[i]);

return 0;

}

**For parser tree(inorder, preorder and postorder traversal):-**

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

struct bin\_tree {

char data;

struct bin\_tree \* right, \* left;

};

typedef struct bin\_tree node;

void insert(node \*\* tree, char val)

{

node \*temp = NULL;

if(!(\*tree))

{

temp = (node \*)malloc(sizeof(node));

temp->left = temp->right = NULL;

temp->data = val;

\*tree = temp;

return;

}

if(val < (\*tree)->data)

{

insert(&(\*tree)->left, val);

}

else if(val > (\*tree)->data)

{

insert(&(\*tree)->right, val);

}

}

void print\_preorder(node \* tree)

{

if (tree)

{

printf("\n%c",tree->data);

print\_preorder(tree->left);

print\_preorder(tree->right);

}

}

void print\_inorder(node \* tree)

{

if (tree)

{

print\_inorder(tree->left);

printf("\n%c",tree->data);

print\_inorder(tree->right);

}

}

void print\_postorder(node \* tree)

{

if (tree)

{

print\_postorder(tree->left);

print\_postorder(tree->right);

printf("\n%c",tree->data);

}

}

void deltree(node \* tree)

{

if (tree)

{

deltree(tree->left);

deltree(tree->right);

free(tree);

}

}

void main()

{

node \*root;

node \*tmp;

root = NULL;

char str1[100];

char newString[10][10];

int i,j,k,temp,m;

char newChar[100];

printf("\nUniversity of petroleum and energy studies.:\n");

printf("\n Input a string for which you want to perform Tockenization : ");

fgets(str1, sizeof str1, stdin);

j=0;

temp=0;

m = 0;

for(i=0;i<=(strlen(str1));i++){

if(str1[i]==' '||str1[i]=='\0')

{

newString[temp][j]='\0';

temp++;

j=0;

}

else

{

newString[temp][j]=str1[i];

newChar[m] = str1[i];

m++;

j++;

}

}

printf("\n tokens after split by space are :\n");

for(i=0;i < temp;i++)

printf(" %s\n",newString[i]);

for(i=0;i < m;i++){

//printf(" %c\n",newChar[i]);

insert(&root, newChar[i]);

}

/\*

insert(&root, 'H');

insert(&root, 'E');

insert(&root, 'L');

insert(&root, 'L');

insert(&root, 'O');

\*/

printf("\nPre Order Display");

print\_preorder(root);

printf("\nIn Order Display");

print\_inorder(root);

printf("\nPost Order Display");

print\_postorder(root);

deltree(root);

}

**WORK DONE**

Till now our project deal’s with the part of tokenization of the text and its hierarchical representation which is according to the updated rules of context free grammar. It is the half part of the project where we can processed the one language through all the measures and in other half part we have to deal with another language with same process but in reversed for language conversion or diversion.

Till now we have one test case with random word set as a sample and algorithm to build the parser tree for the first language texts. Our project is divided in four sections which are described in a methodology part, and every team member works on each and every part. In second part of our project we perform a language conversion after analyzing all the words translate it into a grammar syntax then with the help of parsing techniques we can get our translated language in accordance to the proper syntax of language.

But in second phase we have to make two parser first for manual understanding which is not mandatory, and second for technical purpose which is mandatory because in only this parser we maintain our grammar rules. The code for the same is also completed, basically it is divided into two parts first is tokenization and second is parsing analysis tree, till mid semester we have our completed tokenization code and now we have both codes completed.

**REFERENCES**

https://link.springer.com/chapter/10.1007%2F978-81-322-1143-3\_4

<https://www.semanticscholar.org/paper/Rule-Based-Machine-Translation-from-English-to-Rajan-Sivan/6ea3f88187905e0e6a78ad37d87fc83d49ce1eb8>

<https://fdocuments.in/document/ijcst-vo-l-3-iss-ue-1-jan-m-issn-0976-8491-ncst-mumbai-cdac-pune.html>

<https://link.springer.com/chapter/10.1007%2F978-3-540-30211-7_27>

<http://sanskrit.jnu.ac.in/corpora/JNU-Sanskrit-Tagset.htm>

<https://www.tandfonline.com/doi/full/10.1080/03772063.2018.1528187>

<http://sanskrit.jnu.ac.in/corpora/tagset.jsp>

https://www.researchgate.net/publication/338965896\_Research\_Productivity\_of\_University\_of\_Petroleum\_and\_Energy\_Studies\_during\_2004-2018\_A\_Scientometric\_Analysis