International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 10 | Oct 2019 www.irjet.net p-ISSN: 2395-0072

Analysis of Cocke-Younger-Kasami and Earley Algorithms on Probabilistic Context-Free Grammar

Palash Anjania[1], Shrushti Gangar[2], Krina Bhimani[3], Lynette D'mello[4]

[1,2,3]Student, Dept. of Computer Engineering, Dwarkadas J. Sanghvi College, Maharashtra India
[4]Lynette D'Mello, Professor, Dept. of Computer Engineering, Dwarkadas J. Sanghvi College, Maharashtra India

---

Abstract - *Programming parsers are generally based on context-free grammar. The study of such grammars are of great importance around the world, out of which the CYK and the Earley are the well-known algorithms. In this paper, we have done a comparative study on the above two algorithms. We applied these algorithms on probabilistic context-free grammar (PCFG) and tried to analyses their results and obtained their time complexities. The results obtained show that the Earley algorithm have proved to be better than the CYK algorithm.*

***Key Words*: Cocke-Younger-Kasami algorithm, Earley algorithm, Probabilistic Context-free grammar,**

1. INTRODUCTION

Probability is one of the imperative tools to determine uncertainty. Probabilistic Context-Free grammar uses probability as an addition along with productions. These probabilities can be viewed as parameters for models, for larger models it is advisable to use machine learning techniques to assign these probabilities. PCFG has been a topic of great interest for software developers. CFG is defined as recursive generation of strings from a grammar. PCFG uses probabilities along with CFG i.e. it recursively generates strings with probability in the mix. The set of tuples 'Q' comprises of 5 tuples, Q :{M,T,R,S,P}, where M: Set of non-terminals, T:Set of terminals, R: Set of production rules, S: Start symbol, P: Set of probabilities on production rules.

Probabilistic Context-Free grammar ignores the null in the production, it does not take into consideration the epsilon symbol 'ε' while generating strings. PCFG can be applied in various domains. It has wide applications such as Natural Language Processing (NLP). Parsers are required in many computer field applications where these parsers make use of PCFG, playing a crucial role to solve grammar related problems in computer science. Probabilistic Context-Free grammar also has applications in various other fields like Analysis of Ribonucleic acid (RNA), Protein sequence analysis, Medieval language processing and many more.

2. LITERATURE REVIEW

2.1. CYK Algorithm

2.1.1 Introduction: Cocke-Younger-Kasami algorithm also known CYK or CKY algorithm is a membership algorithm which works only on context-free grammar in Chomsky Normal form (CNF). It is a Bottom-Up Dynamic programming approach and is more data driven. Probabilistic CYK recovers the most probable parse given the probabilities of all productions and used to decide whether the string belongs to the grammar or not. In Context-free Grammar the right-hand side of each production can have either one terminal or two non-terminals.

◦ C→ β

◦ C → EF All possible successive subsequence of letters and sets M ∈ P[i,j] are considered if the sequence of letters starting from a to b can be generated from the non-terminal M. It goes on to sequence of length 2 only if it has considered sequence of length 1, similarly it considers sequence of length 3 after it has considered sequence of length 2 and so on. It considers every possible partition of the subsequence into 2 halves, and checks to see if there is some production X->YZ such that Y matches the first half and Z matches the second half, for subsequences of length 2 and larger. If so, it records X as matching the whole subsequence. Once this process is completed, if the entire string is matched by the start symbol the sentence is recognized by the grammar [3].

2.1.2 Algorithm:

```
1.  Begin
2.      for ( i = 1 to n do )
3.      Vi1 ( A | A → a is a production where ith symbol of x is a )
4.
5.      for ( j = 2 to n do )
6.          for ( i = 1 to n - j + 1 do )
7.          Begin
8.              Vij = ϕ
9.              For k = 1 to j - 1 do
10.             Vij = Vij ∪ { A | A → BC is a production where B is in Vik and C is in
    V(i + k)(j - k) }
11.         End
12. End
```

Fig 1

2.2. Earley Algorithm

2.2.1 Introduction: Earley algorithm can parse any grammar. It is a Top-down Dynamic programming algorithm. For a given production A->αβ, Earley dot

notation which is used to represent current position for a given production can be denoted as A->α • β. A state set is generated for every input position and tuple for each state set is (A → α • β, j) which is divided into three parts. Production used for matching. Position in the right hand's rule side (represented by a dot). Variable 'j' represents origin position. At input position p the state set is called S(p). The parser is seeded with S (0) containing only the higher-level rule. The following operations are iteratively executed by the parser: Prediction: Add (B → • γ, p) to S(p) for every production, for each state in S(p) of the form (A → α • B β, i) where i is the origin, in the grammar with B on the left-hand side (B → γ). Scanning: Add (A → α b• β,i) to S(p+1) for every state in S(p)of the form(A→α• bβ, i), Ifb is the subsequent symbol in the input stream. Completion: Find all states in S(i) of the form (A → α • B β,j) and then add (A →bB • β,j) to S(p)[1], for every state in S(p) of the form (B → γ •, i).

Only new items are added to set in order to avoid duplicity.

2.2.2 Algorithm:

```
function EARLEY-PARSE(words,grammar) returns chart

    ENQUEUE((γ → •S, [0,0]),chart[0])
    for i ← from 0 to LENGTH(words) do
        for each state in chart[i] do
            if INCOMPLETE?(state) and NEXT-CAT(state) is not POS then
                PREDICTOR(state)
            elseif INCOMPLETE?(state) and NEXT-CAT(state) is POS then
                SCANNER(state)
            else
                COMPLETER(state)
        end
    end
    return(chart)
```

Fig 2

3. APPLICATIONS

It is used in stochastic context-free grammar model which is developed for recognizing complex, multitasked activities from Videos and Images. Earley algorithm determines semantic deviation from recognition. Different parsing strategies are introduced in order to enable error detection and recovery in stochastic context-free grammar and also parsing strings allows to recognize patterns. Probabilistic Earley Parser is used as a psycholinguistic model wherein, it is used to calculate to cognitive load. Some other of Earley Algorithm are given below:

•Natural Language Processing: parsing written sentences

•Bioinformatics: RNA sequences

•Stock Markets: model rise/fall of the Dow Jones
•Computer Vision: parsing architectural scenes[8]

4. COMPARATIVE STUDY

Table -1: Comparison table

Factor under consideration	Earley's algorithm	CYK algorithm
Type of grammar	The Earley algorithm can parse all types of grammar.	The CYK algorithm can be used to parse grammar which are in Chomsky Normal Form.
Handling the 'ε'	The Earley algorithm faces many complications when handling grammars containing ε -rules.	The CYK algorithm do not face any such complications.
Grammar rules	In Earley algorithm, words are read one at a time with the help of grammar rules.	In the CYK algorithm, grammar rules are applied on substrings of increasing length.
Time Complexity	The upper bound of earley algorithm is O(n^3), hence the time complexity will be better than this.	O(n^3) is the time complexity for the CYK algorithm.

5. APPLICATION RESULTS:

5.1 CYK Algorithm

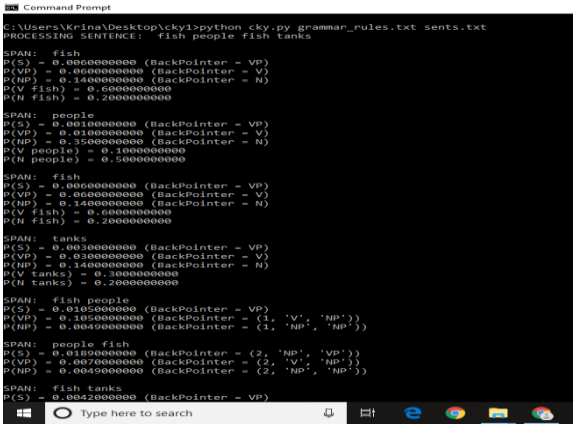

Fig 3

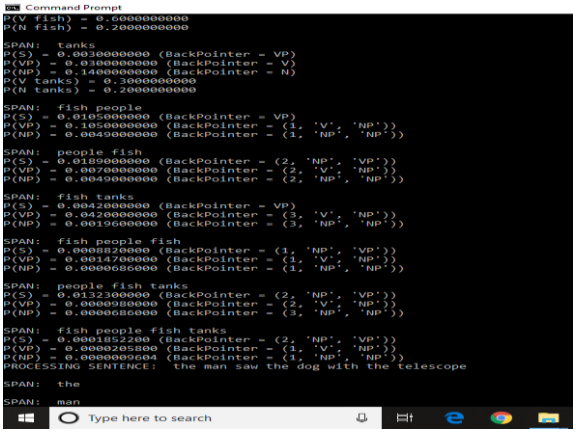

Fig. 4

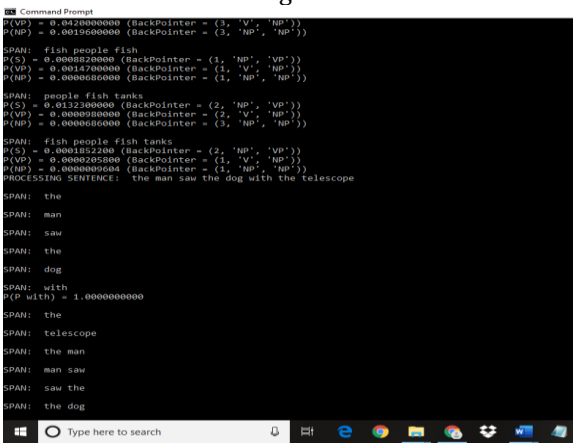

Fig 5

5.1 Earley Algorithm

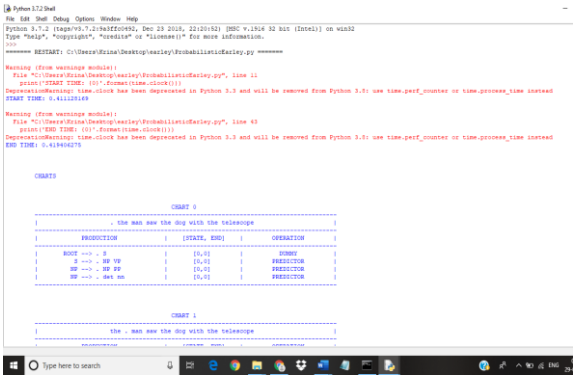

Fig 6

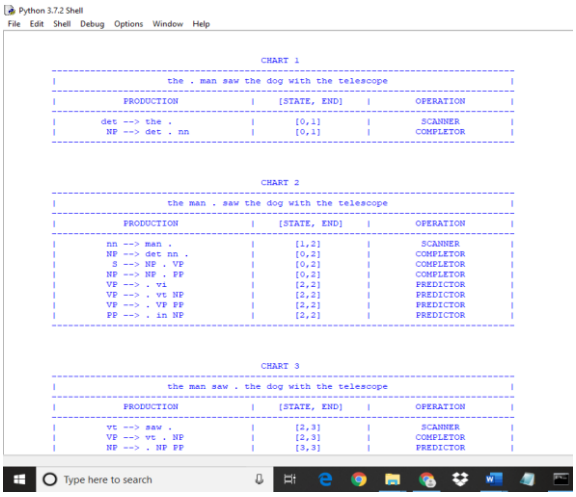

Fig 7

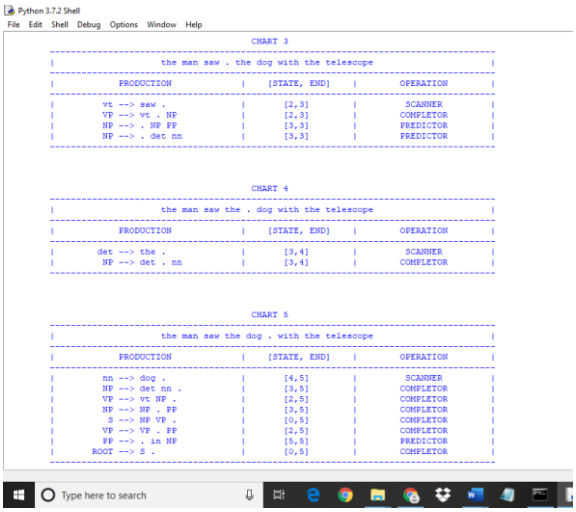

Fig 8

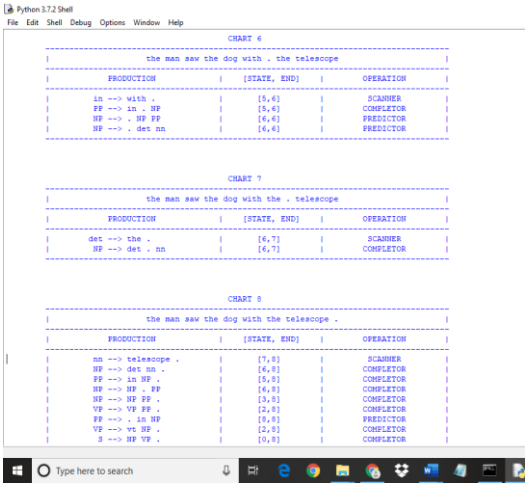

Fig 9

6. CONCLUSION

In the topic of probabilistic context-free grammar, we studied two algorithms namely,. The Cocke-Younger-Kasami and The Earley algorithm. THese algorithms are used in a variety of fields in the industry and they are also used for research in Natural Language Processing , Protein Sequence Analysis and various other applications in the medical domain. The analysis of CYK and Earley algorithm was done using comparative study on a set of input which showed that the Earley algorithm is more efficient than the CYK algorithm in terms of time as well as it can handle grammar is not context-free.

7. ACKNOWLEDGEMENT

This research was bolstered by Prof. Lynette D'mello and Prof. Ramesh Sutar who were involved in guiding us throughout the project.

REFERENCES

1) https://en.wikipedia.org/wiki/Earley_parser

2) https://en.wikipedia.org/wiki/CYK_algorithm

3) https://www.cs.bgu.ac.il/~michaluz/seminar/CKY 1.pdf

4) J. Earley. An Efficient Context-Free Parsing Algorithm. Commun.ACM,13(2):94–102,1970.

5) https://web.cs.ucdavis.edu/~rogaway/classes/120/wi nter12/CYK.pdf

6) [6]PCFGParser,RobertoValenti,0493198,Rómulo Gonçalves, 0536601

7) https://courses.cs.washington.edu/courses/cse401/ 16wi/lectures/10-CYK-Earley-Disambig-wi16.pdf

8) www.cc.gatech.edu › ~phlosoft › files › schindler8803

9) https://www.gatevidyalay.com/cyk-cyk-algorithm/