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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

Optimal Search Algorithm for Text Recognition in Natural Language Processing

Kiran Philip[1], Manimozhi R[2],

Student, Dept. of M.Sc Software Technology, AIMIT, St. Aloysius College (Autonomous), Mangalore, Karnataka, India[1]

Associate Professor, Dept. of MCA, AIMIT, St. Aloysius College (Autonomous), Mangalore, Karnataka, India[2]

ABSTRACT: The parsing technique of Natural Language Processing is done through pattern matching which is related to database. Lot of investigation's has done to reduce the space search for locating same parse of sentence in which different characters can have multiple outcomes and it can be coupled together in different methods. The algorithms contain mechanisms for ordering, which reduces the search cost without any loss of completeness or accuracy. The mechanism that identifies the space and it may result in deleting valid parses for the best parse.

KEYWORDS: Adios, CLL, CDC, POS

I. INTRODUCTION

The Field of Information Technology and computer Science is growing rapidly. The Rapid growth has made total change in the technology and programming paradigm. The Software development these days are very big in size and extremely complex in nature. Natural language is filed of computer science and artificial intelligence that interacts with computer and human languages. The most challenging think in this field is understanding. Modern Natural Language processing are based on machine learning.

The development and benchmarking of the existing and proposed algorithms in the field of NLP are in terms of the accuracy, the search space and the parse time. Speed of an order of magnitude can be achieved without loss of completeness, where decrease of magnitude are achieved in terms of search space. A further order of magnitude reduction of time and search space can be achieved with the help of finding the most probable parse. The sizes of the lexical databases and important grammatical rules shows the behaviour of natural language processing parser. By increasing the database size or by including some complex set of grammatical rules and the parsers are able to handle the parsing of more complex sentences [1]. Without the database or rules, parsing of big sentences are avoided due to the extremely large amount of different possibilities in parsing the sentence [1]. Increase in the parse time form the application of complex grammatical rules, we implement a search algorithms to a parser to reduce the search space and parsing speed. To measure the accuracy of the parser, we implemented a scoring system [1] This scoring system are derived from the probability that a particular structure would exist and it does not always parse the sentence correctly [1], but provides a good indication of the structure from a statistical point view.

II. LITERATURE REFERENCE AND WORKS

The main research studies in the field of natural language processing have been made in various application fields such as speech recognition (Baker 1979), computational linguistics (Adrian's 1992), computational biology (Sakakibara et al. 1994; Salvador and Benedı 2002), and machine learning (Sakakibara 1997; de la Higuera and Oncina 2003). The majority of grammars which are developed for natural language has been implemented based on context-free grammars alternative of context-sensitive ones.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

1. ADIOS

The ADIOS (Automatic Distillation of Structure) algorithm was proposed as a natural language processing method of grammar induction that focus on symbolic results which are in the form of a context-free grammar . It induces grammars such as text, transcribed speech, and nucleotide base pairs which uses positive examples in an unsupervised fashion. Because of unsupervised faction ADIOS algorithm is classified into a text-based grammatical method and unsupervised grammatical inference method.

2. CLL

The CLL algorithm aims to learn natural language syntax from a corpus of declarative sentences and without the need of an oracle for validating hypotheses about the language which are used. Therefor it belongs to the class of *text-based* grammatical inference method and *unsupervised* grammatical inference method.

CLL is not only considered with developing a feasible language learner but one that is also psychologically plausible too. CLL is trying to emulate a child with respect to its acquisition of its first language

3. CDC

This algorithm was introduced for the unsupervised induction of context-free grammars from tagged text by Clark. Commonly, CDC can be classified into a *text-based* and *unsupervised* grammatical inference method. This algorithm makes use of two techniques: one is distributional clustering and the other is mutual information. This technique is used for identifying sets of common sequences which are derived from non-terminal.

III. NEW FRAMEWORK AND IMPLEMENTAION METHOD

The purpose of the algorithm is to provide a faster way of parsing sentences without losing the effect of grammatical structure, or the semantic and syntactic information that have been extracted from the parser [1]. These are the key areas being focus of most research done in NLP and will continue to increase in complexity in the future [1].

A. Parsing

Parsing also called syntactic analysis is the process of identifying strings or symbols not only in computer language but also in natural language processing. It is a software component that takes input data and builds.

The training stage, the parser builds up the grammatical structure model by identifying from a manually parsed corpus. The CCG2 in the parser consist of certain rules and methods which are used for combination stage of the parser, and implements s set of the standard CCG combinatory that makes the grammar more flexible.

Initially, the training corpus for the particular word are given a set of categories based on words. Due to the changes in the training data, a huge set of potential categories due to varieties in the training data are given for some words. If the search word was not found in the training corpus, a lexical databases also called Word Net is used. This can be achieved by extracting the part of speech for the unknown word and assigning all the possible categories to the word for the POS. These states are derived from three combination of transition probabilities. They are categorical transition, the lexical transition and word transition

B.Optimal search algorithm

The major goal of this project was to introduce alternative standard and novel algorithms. The algorithm first considered was Adaptive probing. This algorithm was considered due to speed of searching that seen in the search problem, but it was rejected due to the random nature of the searching algorithm. The first enhancement was to apply a different ordering of the combinations to allow the fast build-up of the relevant sections of the parse tree [3]. By ranking the states in order of their probabilities, the parse tree was built up in such a way that the most probable state in the tree was considered first [3]. Due to the randomized build-up of the parse tree in terms of extension of the branches

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

in the search tree, the algorithm had to include an indicator to allow the extension of branches from nodes, even after it had been used to construct its children states already [3]. This backtracking mechanism was implemented using a list some states, which was divided into two parts. The pointer indicates the division point between the two parts, which contains the states that used for combining with other states, and the other section which contains the states that are not used combine with remaining states. Another major constraint of the first algorithm is that it often ignored, which is the overhead at the time of execution. This idea plays an important role in the search problem, but it was ignored due to the increase in rate of hardware performance. The algorithmic design was modularized, because of that it provides an easy switching of the algorithm with the original one.

The first algorithm considered was Adaptive probing and was tested on the basis of subset of the problem by using a toy language. This algorithm was considered due to the gain in search speed seen in the simplified search problem, but was rejected due mainly to the random nature of the search, which means that an exhaustive search was necessary to provide the most probable parse[3]. The main thing to do is to apply a different order of combinations that allows the fast building of the important sections of the parse tree. This backtracking mechanism was implemented with the help of states, which was divided into following sessions. A pointer points the division point between the two parts, first part contained all of the states that had been used for combining with other states, and the other part contained all the states that are not used for combining with other states [1].

Ranking Algorithm

A simple algorithm for maintaining the ordered list was introduced and is as follows.

1. Identify the list for ealvery word with every state
2. Sort the list according to the scores
3. With the help of list set the pointer at the first state.
4. While the list contains un-combined states:
5. Set pointer as the next most probable state.
6. Return if pointer state is a terminal state.
7. Combine pointer with higher probability with all adjacent states.
8. Use Insertion sort to all newly created states into the list.
9. Return failure

The scores received for the most probable states of each word are used to derive the normalization scores for the particular sequence sequences. This was not the most accurate way of determining the normalization scores, but it provided an efficient way to change the order of pre-processing stage and is derived by,

$$S_{i,j}^{normal} = \prod_{k=i}^{j} S_{k,k}^{non}$$

$$S_{i,j}^{rank} = S_{0,i-1}^{normal} \cdot S_{i,j} \cdot S_{j+1,J}^{normal}$$

$$= S_{0,J}^{normal} \cdot S_{i,j} / S_{i,j}^{normal}$$

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

i and *j* represents the starting and ending indexes of the state and *l* indicates the length of the sentence. *Si,j normal* represents the normalization score for the sequence *Sk,k max* represents the score of the most probable states for word at *k*. *Si,j rank* represents the score used for ranking, but it also represents the heuristically score of the state. *Si,j* represents the probability score of the particular sequence.

Combined algorithm

The combined algorithm maintains the selection of the parse tree with the similar probabilistic score but the difference is that it has the ability to manage very large section of the search tree without creating too much load in the execution of the algorithm. The new algorithm is as follows:

1. Create the normalization mapping.
2. Let the critical score to zero.
3. Sort the sorted table with all states by using the normalization score
4. Remove states that are not needed and insert those states into the indexed table.
5. While the sorted table contains not combined states:
6. Remove the most important state from the sorted table as the pivot.
7. Return when the pivot is a terminal state.
8. Combine pivot with all adjacent states in the indexed table and check whether and scores we received don't fall below the critical score.
9. While every state that has been created:
10. If the project score is terminal state adjust the critical score.
11. With the normalized score insert the created state into sorted table
12. Add the pivot into the indexed table.
13. Return failure.

This is an alternative algorithms that included more parsing in the search tree, and also the effects of prematurely ending the search when same result was found. The ideas such as pruning lower scored states at the start of the algorithm (beam search), approximating the correct parse to be the first terminal state it found etc. are tried to implement in it. The beam search has the same effects to the parser as a reduced set of categories and combinatory, in that, some valid sentences could not be parsed because of the reduced amount of ways in forming the valid parse[1]. This is a very common approach used in NLP to optimize a search keyword, but was not accepted this approach for this project because the task of this algorithm is to find the probable parse for the sentences.

1V. CONCLUSION AND FUTURE WORKS

Unlike all modern search algorithms which will takes advantage of increasing processing power of the modern day computers and hence result in loss of search technique, which developed for search algorithm for the retrieval of the best possible solution in a very efficient manner.

The implementation of the algorithm have searching mechanism to find the most parsers for the target parser has that reduced the parsing time required to retrieve the similar result. The features of this algorithm has the potential to convert into a simple parser, which are helpful for extracting the relevant information.

Our endeavour stands incomplete if we do not dedicate our gratitude to all who gave valuable contribution towards the presentation of this research paper. First and foremost, we thank God Almighty, for his kind blessings showered on us. We express sincere gratitude to our Dean and HoD of the IT Department.

REFERENCES

[1] Application of search algorithms to natural language processing (2003) - Takeshi Matsumoto
[2] A survey of grammatical inference methods for natural language learning - Arianna_DUlizia