SELECTED RESEARCH PROBLEMS IN THEORETICAL COMPUTER SCIENCE

[1]Gend Lal Prajapati
Department of Computer Engineering, Institute of Engineering and Technology
Devi Ahilya University, Indore
Email:[1]glprajapati1@gmail.com

Abstract— Working in the core areas of computer science has always been difficult, especially for beginners. Difficulty arises in finding a research problem and to understand the research process, as one should have complete understanding of the field in order to pursue research. In this paper, we present some selected research problems in theoretical computer science those having much practical use. This paper is to generate the research interest in beginners to work in this field. After going through the paper readers will be able to bank their research objective, ideas to pursue, and important articles for creating their basic foundations. We also present examples for better understanding, and directions to position the research work.

Index Terms—Theoretical Computer Science, Formal Languages, Lazy Learning, N-gram, Machine Translation.

I. INTRODUCTION

Research in theoretical computer science mainly deals with the study of mathematical concepts and concerns in obtaining computational models for the formulation of problems. It also derives its motivation from practical aspects and computation. Research results require mathematical proof and rigorous analysis in order to support its validity and soundness. Accuracy of research results requires to go for implementation of abstract model with benchmark data sets in order to see how results compare with others. After the invention of Turing machine algorithms are considered as a set of computational steps. Thus, the research process in this area can be viewed as to give an algorithmic model of computation for the problem under study; then we need to show it correctness, analysis and finally we should go for profiling works for the confirmation of theoretical analysis and positioning the work.

Theoretical computer science is a broad field within general computer science. Research areas like algorithms, data structures, language theory, automata theory, parallel computation, Computational biology, machine learning, algorithmic learning theory, randomization, distributed computation, very large scale integration, computation complexity theory, etc. are regarded to fall in theoretical computer science field. The problems generally in this field are seemed to be computationally hard but interesting. Formulation of such problems requires having sound knowledge in the area.

In this paper, we discuss some notable problems that can be considered as a line of work. This will be useful for beginners those wish to peruse PhD (or research) and are in search of the problems to take as a research project.

II. LEARNING OF FORMAL LANGUAGES

Learning of formal languages, often called grammatical inference, is an active research area in machine learning and computational learning theory. In machine learning machine seems to be learn by experience. The problem of learning a correct grammar for the unknown

language from finite examples of the language is called grammatical inference. In particular, the problems of learning finite automata from positive and negative examples for the case of regular languages have been widely and well studied. Inductive learning of formal languages inherently contains computationally hard problems. For example, the problem of finding a deterministic finite automaton with a minimum number of states consistent with a given finite set of positive and negative examples is known as NP-hard [1]. The first convincing model for the case grammatical inference is introduced by Gold in 1967 [2]

The problem of inductively learning of context-free grammars from positive and negative examples is a more difficult learning problem than learning finite automata.

A grammar is a quadruple $G = (N, \Sigma, P, S)$, where N and Σ are the alphabets of non-terminals and terminals, respectively, such that $N \cap \Sigma = \emptyset$. P is a finite set of productions and $S \in N$ is a special non-terminal called the *start symbol*. A grammar G is called *context-free* if all production rules are of the form $A \rightarrow \alpha$, where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$. In the production rule $A \rightarrow \alpha$, A is called as the left-hand side (LHS), and α is the right-hand side (RHS). A language L is a context-free language if there exists a context-free grammar (CFG) G such that $L = L(G)$.

If we are given plain text (i.e., positive examples of the language) and asked to determine the grammar consistent with the text then it is a difficult problem. As in the search space of grammars there are possibly infinite grammars. We need to exhaustively find all possible grammars and select that is consistent with input text. The difficulty arises from two specific aspects of the problem of learning context-free grammars from examples: first determining the grammatical structure (topology) of the unknown grammar, and second identifying the set of non-terminals in the grammar. The first problem is especially hard because the number of all possible grammatical structures for a given positive example becomes exponential in the size of positive examples. Thus, the hypothesis space of context-free grammars is very large—too large in which to search for a correct context-free grammar consistent with the given examples. Sakakibara has shown that if information on the grammatical structure of the unknown context-free grammar is available for the learning algorithm, there exists an efficient algorithm for learning context-free grammars from only positive examples [3]. Otherwise there is no known polynomial time algorithm available for learning context-free grammars from plain text.

Grammatical Inference has its wide applications including Computational Biology. Natural language processing, learning programming languages, information extraction, pattern recognition, etc.

Difficulty of learning context-free grammars has given rise different methods including learning tree automata, artificial intelligence like approach using genetic algorithm, and using fuzzy similarity measure.

A. Learning Tree Automata

A **tree automaton** [3] is a type of state machine which deals with tree structures, rather than the strings (or words). On input structural examples, Sakakibara first constructed tree automata then using tree automata context-free grammar are constructed. Running time of this algorithm is polynomial in terms of the size of input skeletons (structural examples) [3].

B. Context-Free Grammars Learning Using Genetic Algorithm

A genetic algorithm is a guided random search technique used to find near optimum solution. Solution obtain using genetic algorithms may or may not be optimal. This technique seems better when search space is very large and brute force techniques fail to apply. Genetic algorithms use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. Starting from the set of initial solution we go generation by generation in order to find the better solution.

Sakakibara and Kondo have used genetic algorithms (GAs) to search for a context-free grammar in Chomsky normal form, consistent with the given examples [4]. Experiments suggest that the knowledge of part of the structure (some parenthesis) may help and reduce the number of generations needed to identify the correct grammar and thus contribute to improving the efficiency of the learning algorithm [5].

C. Context-Free Grammars Learning Using Fuzzy Similarity Measure

This is an alignment based learning. *Alignment based learning* (ABL) [6] is based on the alignment of sentences and substitutability. In the alignment phase the matched parts of sentences considered as possible constituents. Equal parts and unequal parts are used for generating context-free grammar production rules. Non-terminals are assigned as the left hand sides of the production rules with the unequal parts as the right hand sides shown by the following example:

[Ram] sees the [green] apple.
[Shyam] sees the [red] apple.

The created grammar rules are:

S → NT_A sees the NT_B apple,
NT_A → Ram, NT_A → Shyam
NT_B → green, NT_B → red

In particular, alignment profile is generated to obtain the consistent grammar [7]. However, learning context-free grammars from positive examples without using any heuristic information is still a challenging problem.

III. SEARCH IN HIGH DIMENSIONS

Nearest Neighbor (NN) search in high dimensional data is widely used method which has applications for image/information retrieval, medical imaging, DNA matching, pattern recognition, machine learning, data mining and many more. Search for high dimensional data/query in feature space is still a dimensionality curse [8] where probability of finding nearest neighbor decreases as number of dimension grows.

Accuracy of result is based on important factors like memory requirement, number of dimensions, clustering algorithm and many more. Further research is required in area of exact NN search methods which consider all features of data.

IV. CLASSIFICATION PROBLEM

Support vector machine (SVM) is applicable in many domains for classification. SVM classification is the main approach after that many researchers divert the path for accuracy and efficiency in SVM classification from using the analyzing features and kernel [9].

Many research done in SVM classification with selected features and kernel in different domain area; but some are related to feature, some are related to kernel and some are related to any particular application domain. Designing an efficient classifier is still an active research problem.

V. QUERY OPTIMIZATION TECHNIQUES

Query Optimization is very important function of Database Management System (DBMS). Query Optimizer translates user-submitted Query into different Query Evaluation Plans (QEPs). All plans are equivalent in terms of their final output but vary in their cost, i.e. the amount of time that they need to run. Query Optimizer explores different plans and attempts to determine most efficient plan for executing the query. The Process of creating, exploring and selecting an efficient plan is called Query Optimization. The Query Optimizer in standard DBMS picks a single Plan and executes Query with the selected plan. The chosen plan aims to minimize running time by carefully optimizing the use of secondary storage, memory, and CPU [10].

Now days Databases are growing exponentially with increasing complexity to manage wide variety of local and global Data, static and dynamic Queries, large datasets, large number of operators. More statistics, more metadata required to handle so many number of tables, relations, attributes. An optimizer is required to select an efficient plan among thousands of plans. Moreover, plans at optimization and execution time increase and hence computational cost increases as well as performance of an optimizer may not be optimum. Therefore obtaining an efficient optimizer is still a challenge.

Now scientists have gone to Adaptive query processing technique as it focuses on using runtime feedback to modify query processing in a way that provides better response time or more efficient CPU utilization [11]. Several Adaptive Query Processing Techniques have been developed to address the query processing issues [12]. Adaptive Query Processing techniques still not capable of handling complete issues so other query processing techniques including obtaining information not stored in the catalog and providing it just-in-time to the optimizer [13],

learning from past queries and mistakes [14], Clustering and using techniques from data stream systems can be combined together to reduce query processing problems [10]. Thus beginners may take this line of work in order to obtain an efficient query processing scheme.

VI. LAZY LEARNING APPROACHES

Lazy learning approach for classification also provides ample rooms for research. Classification is one of the important techniques of supervised learning, which is used to predict categorical class labels. It mainly classifies the data (constructs model) on the basis of data provided for training with class labels and uses it to classify new data. Many effective classification algorithms have been developed such as Decision tree, Naive Bayes, Neural networks and so on.

In general a classifier may generate large number of rules satisfying some quality constraints. However, during classification many of these generated rules may be useless and worst. Lazy (non-eager) learning classification overcomes this problem by focusing on the features that actually occur within the testing instances. Lazy learning classifier is to establish a classification of the rules for features that actually occur within the testing instances and to increase the chance of generating more rules that are useful for classifying the test instance. In literature, we can see different lazy learning approaches [15]. As a future research one can consider to work on improving computing efficiency of Lazy learning classifiers, for which the computation is performed on a demand driven basis.

VII. BIOLOGICAL COMPUTATION

Since the invention of Turing machine biological scientists have started using computational model for their biological computation. In this field there are lots of opportunities for research. There is still required to develop efficient tools/techniques for automatic DNA sequencing, protein structure prediction, obtaining phylogenetic tree, visualization tools, sequence analysis, etc.

Biological data are large in size, we need to deal with very large size sequences of DNA, RNA and protein, and therefore the problems in this area are generally found as computationally hard. Determining similarity between two sequences (i.e., obtaining pair wise alignment), among a family of sequences (i.e., multiple sequence alignment), classifying the members in different families according to their evolutionary distances and discovering the right ancestor of a candidate member will continue to be some of the most important and fundamental computational tasks. Some basic techniques for pair wise alignment can be seen in [16], [17]. Wang *et al.* describe techniques for DNA sequence classification tasks in [18].

VIII. AREA OF LANGUAGE PROCESSING

Language processing is a broad area that includes natural language processing, computational linguistics, and speech recognition, etc. This field always has been challenging for the scientists to work. Most of the time is spent for dealing with ambiguities. For example, the sentence “go to the bank” has ambiguity. Bank can be a financial bank, bank can be a river bank, bank can be a noun, bank can be a verb, etc. We expect the most probable meaning of this sentence which can be resolve by a particular context, past experience etc. machine needs to consider all possible contexts which may be infinite. Also past experience (database) should be processed as large as possible. The accuracy increases if we could process all possible contexts as well as reasonably large database. Therefore, for such computation computer requires large memory space and CPU time. This suggests that language processing seems to out of our practical reach. We need to determine tradeoff between accuracy and resource requirements.

There are good numbers of topics to choose as a line of work including resolving the ambiguity, parts of speech tagging, correcting spelling errors, determining the semantics, discourse analysis which aims to determine the meaning of group of sentences instead of individual sentence or word processing, and speech recognition.

Guessing the next most probable word is a crucial subtask of speech recognition, and hand-writing recognition etc. There are number

of language models can be seen in the literature including n-gram [19], [20].

The n-gram language model estimates the probability of a word, given previous words ($w_n|w_1, \ldots, w_{n-1}$) where n is usually set to 2 (bigram), or 3 (trigram). The literal meaning of the word gram is token. The n-gram is a sequence of token in a given sentence. It is used to estimate the probability of a word occurring in n-1 context. It is used to model the syntax and semantics of natural language. An n-gram model relies on a Markov assumption that each word depends only on n-1 previous word. This works well with small training data. But, as training data size increases, language model size also increases, and then it becomes difficult to model the data for practical use.

A line of work is to find efficient language models for the word prediction and to evaluate their performance in order to positioning our work. To evaluate the performance of language models, we have various measures like Perplexity (PP), Word-Error-Rate (WER), and Relative Entropy (RE) [21].

Another important aspect in n-gram language modeling is smoothing to avoid zero probability estimations for unseen data. Various methods for smoothing techniques are developed so far. An interaction between pruning and smoothing can be seen in [22]. Further work is still required in the field of smoothing algorithm. In following we discuss two related areas that also seem to be promising.

A. Information Retrieval

Day by day knowledge (information) is increasing in exponential manner. Thus efficient management of such enormous documents becomes challenging. This makes the attention of research scientists to device scheme for storage of information and their subsequent retrieval. Information retrieval system depends on the way information is stored. It can be general, or for some specialized kinds of information. Vector space model is one of the information retrieval schemes where documents and queries are represented as vectors of features [23]. Features consist of terms that occur within the documents, with the value of each feature about presence or absence of given terms in a given document. So, information retrieval can be viewed as an optimization problem where objective is to obtain the document with respect to a given query where maximum numbers of terms are in common.

For evaluating an information retrieval system we need to know the notion of terms Recall and Precision [24]. These metrics we need to maximize. Recall is used to see the ability of system to retrieve relevant documents as follows:

$$\text{Re}call = \frac{No.\,of\ relevant\ \ documents\ \ returned}{Total\ no.\,of\ \ relevant\ \ documents\ \ in\ the\ col\,lection}$$

Precision is a measure of accuracy of the system as given below:

$$\text{Pr}ecision = \frac{No.\,of\ \ relevant\ documents\ returned}{No.\,of\ \ documents\ returned}$$

B. Machine Translation

The problem of translating from one language to another is known as machine translation (MT). We may require going for some or full process of MT depends on our need and application. If we have formal models for MT then many real problem including communication gap due to language barrier may be resolved. However, achieving full MT seems to be hard [24]. The MT process depends on the syntactical, morphological, semantic, and pragmatic structures of a particular language. So, we actually require developing efficient algorithms and data structures for handling the issues of natural languages. There are numbers of contributions have been done for MT using different methods including statistical approach. Some approaches can be seen in [25], [26], and [27].

IX. CONCLUSION

We have presented some selected research areas in theoretical computer science that we feel primarily important. We have also provided important literatures that will be very useful for beginners to understand the field and to find the nature work needed in this area.

REFERENCES

[1] Gold, E.M., "Complexity of Automaton Identification from Given Data,"

Information and Control, vol. 37, pp. 302–320, 1978.

[2] Gold, E. M., "Language Identification in the Limit," *Information and Control*, vol. 10**,** pp. 447-474, 1967.

[3] Sakakibara, Y., "Efficient Learning of Context-Free Grammars from Positive Structural Examples," *Information and Computation*, vol. 97, pp. 23-60, 1992.

[4] Sakakibara, Y., and Kondo, M., "GA-Based Learning of Context-Free Grammars Using Tabular Representations," In: *Proceedings of 16th International Conf. Machine Learning*, pp. 354–360, 1999.

[5] Sakakibara, Y., and Muramatsu, H., "Learning Context-Free Grammars from Partially Structured Examples," in: Oliveira, A. de (Ed.), *Grammatical Inference: Algorithms and Applications,Proceedings of ICGI '00*, Lectures Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, pp. 229–240, 2000.

[6] M. V. Zaanen, "Theoretical and practical experiences with alignment based learning," presented at the Australas. Lang. Technol. Workshop, Melbourne, Australia, 2003.

[7] G. L. Prajapati,and N. S. Chaudhari," Learning Alignment Profiles for Structural Similarity Measure". 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA).

[8] Y. Tao, K. Yi, C. Sheng, and P. Kalnis. "Quality and Efficiency in high dimensional nearest neighbor search." ACM Symposium on Information Gathering and Management of Data (SIGMOD). pp. 563–576. 2009.

[9] L. Mangasarian and W. Wild, Feature Selection for Nonlinear Kernel Support Vector Machines, Seventh IEEE International Conference on Data Mining, 2007.

[10] Pedro G. Bizarro Adaptive Query Processing. Dealing with incomplete and uncertain statistics. University of Wisconsin-Madison, 2006.

[11] Amol Deshpande, Zachary Ives and Vijayshankar Adaptive Query Processing. Raman Foundations and Trends in Databases Vol. 1, No. 1 (2007) 1–140, 2007.

[12] J. M. Hellerstein, M. J. Franklin, et al. Adaptive query processing: Technology in evolution. *IEEE Data Engineering Bulletin*, 23(2):7–18, June 2000.

[13] B. Babcock and S. Chaudhuri. Towards a Robust Query Optimizer: A Principled and Practical Approach. In *Proc. of the ACM Intl. Conf. on Management of Data (SIGMOD'2005)*, June 2005.

[14] M. Stillger, G. Lohman, V. Markl, and M. Kandil, "LEO – DB2's Learning Optimizer," in *VLDB '01: Proceedings of 27th International Conference on Very Large Data Bases*, Morgan Kaufmann, September 11–14 2001.

[15] Syed Ibrahim. S.P., Chandran.K.R, Nataraj.R.V, "LLAC: Lazy Learning in Associative Classification" in the Springer Lecture Series in Communications in Computer and Information Science (CCIS), Advances in Communication and Computers, 190, Part I, pp. 631–638, 2011.

[16] Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) "Basic Local Alignment Search Tool" J. Mol. Biol., 215, 403–410.

[17] Needleman, Saul B. and Wunsch, Christian D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins" Journal of Molecular Biology 48 (3): 443–53.

[18] Wang, J.T. L., Rozen, S., Shapiro, B. A., Shasha, D., Wang, Z., and Yin, M., "New Techniques for DNA Sequence Classification," Journal of Computational Biology, Vol. 6(2), pp. 209-218, 1999.

[19] B.F. Brown, P.V.deSouza, "Class-based n-gram models of Natural Languages," Association of Computer Linguistics, vol. 18, no. 4, pp. 467-479, Dec 1992.

[20] Z. Su, Q. Yang, Y. Lu, and H. Zhang, "A Prediction System for web Requests using n-gram Sequence Models," *IEEE Computer society, pp. 214-221 ,2000.*

[21] *I.Oparin, M. Sundermeyer, H. Ney, J. Gauvain, "Performance analysis of neural networks in combination with n-gram language models," ICASSP*, pp. 5005-5008, 2012.

[22] V.TeemuHirsimaki and S. Virpoija , "On growing and pruning kneser-Ney Smoothed n-gram Models," IEEE Trans. on audio, speech, and language processing, vol. 17, no. 1, pp. *1617-1624,* January 2009.

[23] Salton, G., The SMART retrieval system: Experiments in Automatic Document

Processing. Prentice-Hall, Upper Saddle River, NJ. 1971.

[24]Daniel Jurafsky,and James H. Martin, *Speech and language Processing: An introduction to natural Language Processing, Computational Linguistics, and Speech Recognition*, Pearson Education Inc., 2008.

[25]Lopez, A., "Statistical machine translation," ACM Computing Surveys, Vol. 40, No. 3, Article 8, 2008.

[26]Yetunde O. Folajimi and Omonayin Isaac, "Using Statistical Machine Translation (SMT) as a Language Translation Tool for Understanding Yoruba Language, "EIE's 2nd Intl' Conf.Comp., Energy, Net., Robotics and Telecom, 2012.

[27]Yizhao Ni, Craig Saunders, Sandor Szedmak, amd Mahesan Niranjan, "Exploitation of Machine Learning Techniques in Modelling Phrase Movements for Machine Translation," Journal of Machine Learning Research, 12, pp.1-30, 2011.