

Using A* Algorithm in Repeated Acquisition from Similar Websites

Mr. P. Praveen Chandar^{#1}, Miss. J. Ramya Rajalakshmi^{#2}, Mr. S. Mugilan^{#2}

^{#1}Master of Technology, Department of Computer Science & Engineering,

Dr. S.J.S Paul Memorial College of Engineering and Technology, Puducherry – 605 502

^{#2}Master of Technology, Department of Computer Science & Engineering,

Dr. S.J.S Paul Memorial College of Engineering and Technology, Puducherry – 605 502

^{#2}Master of computer application, Department of Master of computer application,

Sri Manakula Vinayagar Engineering College is an engineering college in Pondicherry– 605 107

Abstract -In this paper we discuss about the Knowledge of the similar websites with the help of rule ontology in repeated acquisition with A* algorithm. The ontology can reduce the amount of information and reduce the exertion of utilizing the information in rule acquisition, because it is generalized and specifically reorganized for rule acquisition. We expect that it will be easier to acquire rules from a site by using similar rules of other sites in the same domain rather than starting from scratch. This algorithm will enhance the rules which are in combined form of web sites into a single batch to provide an effective retrieval of rules.

Key Terms: Ontology, RuleToOnto, A* Algorithm, Rule Acquisition, Best First Search.

1. Introduction

ONTOLOGY is an important foundation of knowledge that represents the real world. From the perspective of computer science, ontology defines a set of representational primitives with which one models a domain of knowledge or discourse. These representational primitives are able to be comprehended by human beings and come with machine-readable formats that are composed of the classes, the attributes (properties), and the relationships between classes. A well-constructed ontology can help develop a knowledge-based information search and management system more effectively, such as Web search engine semiautomatic text classification and content management system.

In addition, recent studies also reflect that the ontology-based induction can indeed obtain effective and outstanding performances in many real applications, e.g., risk management in e-Business and e-Commerce task, interoperability between enterprises, multimedia integrated service system,

multiagent e-Learning system, telecom operator service, etc. To sum up, the ontology can be used as the standard knowledge representation for the semantic Web-based systems. Ontology is widely recognized as an appropriate knowledge representation technology; therefore, research on ontology is becoming more in demand for developing knowledge-based information systems.

2. Ontology-based rule acquisition procedure with genetic algorithm

In this section, we recommend a process which routinely acquires rules through RuleToOnto.

In step 1, RuleToOnto is produced from rules which are acquired in another site.

In step 2, variables and values are routinely recognized as of the Web page by means of RuleToOnto and the first rule draft is caused.

In step 3, rules are automatically invented by combining the identified variables and values. We developed A* algorithm for this purpose. Still, the generated rules may be unfinished. So, the knowledge engineer requires to filter the second rule draft to make it complete.

In step 4. Rule Ontology Generation RuleToOnto is domain specific knowledge that provides information about rule components and organizations. It is feasible to directly use the rules of the previous system as an alternative of the proposed ontology. Conversely, it requires a large space and extra processes to utilize information on rules, while RuleToOnto is a generalized compact set of information for rule acquisition. Thus, we use RuleToOnto instead of the rules themselves. While the rule component identification step needs variables, values, and the relationship between them, the rule composition step needs simplified rule structures. Therefore, RuleToOnto represents the IF and THEN parts of each rule by attaching rules with variables with the IF and THEN relations, in addition to essential information about variables, values, and

connections between variables and values. The RuleToOnto representation has three object properties has Value, IF and THEN, and three classes, Variable, Value, and Rule, which is an RDF graph produced from the OWL ontology by the RDF validator.

3. Ontology-based rule acquisition procedure with A* algorithm

In computer science, A* (pronounced "A star") is a computer algorithm that is widely used in path finding and graph traversal, the process of plotting an efficiently traversable path between points, called nodes. Noted for its performance and accuracy, it enjoys widespread use. (However, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance.

4. Description

A* uses a best-first search and finds a least-cost path from a given initial node to one goal node (out of one or more possible goals). As A* traverses the graph, it follows a path of the lowest expected total cost or distance, keeping a sorted priority queue of alternate path segments along the way.

It uses a knowledge-plus-heuristic cost function of node (usually denoted) to determine the order in which the search visits nodes in the tree. The cost function is a sum of two functions:

I. The past path-cost function, which is the known distance from the starting node to the current node (usually denoted)

II. A future path-cost function, which is an admissible "heuristic estimate" of the distance from to the goal (usually denoted).

The part of the function must be an admissible heuristic; that is, it must not overestimate the distance to the goal. Thus, for an application like routing, might represent the straight-line distance to the goal, since that is physically the smallest possible distance between any two points or nodes.

If the heuristic h satisfies the additional condition for every edge x, y of the graph (where d denotes the length of that edge), then h is called monotone, or consistent. In such a case, A* can be implemented more efficiently—roughly speaking, no node needs to be processed more than once (see closed set below)—and A* is equivalent to running Dijkstra's algorithm with the reduced cost.

5. Implementation and Evaluation of A* Algorithm

In this section, we recommend a process which routinely acquires rules through RuleToOnto.

In step 1, RuleToOnto is produced from rules which are acquired in another site.

In step 2, variables and values are routinely recognized as of the Web page by means of RuleToOnto and the first rule draft is caused.

In step 3, rules are automatically invented by combining the identified variables and values. We developed A* algorithm for this purpose.

In step 4. After using A* algorithm all the rules are grouped into a batch. The user will extract the rules for similar websites from the batch. It reduces the extraction of each rule at every time with longer time and produce approximate results. The A* algorithm will enhances with the effective usage of the extraction of rule in the file.

function A*(start,goal)

closedset := the empty set//**The set of nodes already evaluated.**

openset := {start}//**The set of tentative nodes to be evaluated, initially containing the start node.**

Came_from: = the empty map//**The map of navigated nodes.**

g_score [start]:= 0//**Cost from start along best known path.**

// **Estimated total cost from start to goal through**

y.f_score[start]:=g_score[start]+heuristic_cost_estimate(start, goal)

while openset is not empty current := the node in openset having the lowest f_score[] value

if current = goal

return reconstruct_path(came_from, goal) remove current from openset add current to closedset

for each neighbor in neighbor_nodes(current)

tentative_g_score := g_score[current] + dist_between(current,neighbor)

if neighbor in closedset

if tentative_g_score >= g_score[neighbor]

continue

if neighbor not in openset or tentative_g_score < g_score[neighbor]

came_from[neighbor] := current

g_score[neighbor] := tentative_g_score

f_score[neighbor] := g_score[neighbor] + heuristic_cost_estimate(neighbor, goal)

if neighbor not in openset

add neighbor to openset

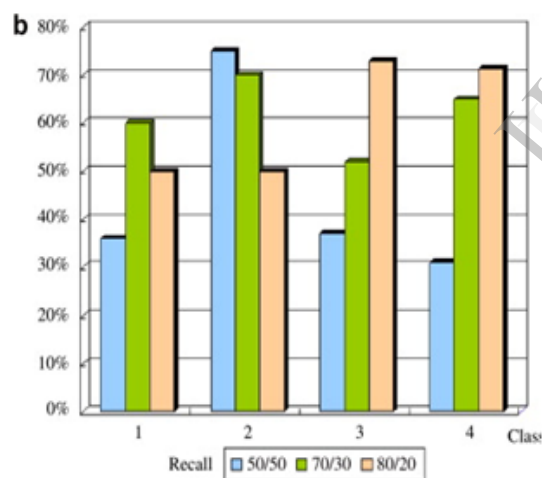
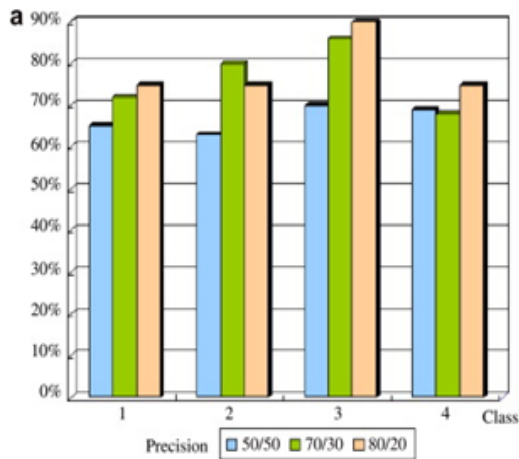
return failure

```

function      reconstruct_path(came_from,
current_node)
if came_from[current_node] in set
    p      :=      reconstruct_path(came_from,
came_from[current_node])
    return (p + current_node)
else
    return current_node

```

6. Precision and Recall Rates of A* Algorithm:



7. Conclusion

This paper has described a comprehensive and innovative ontology-based system framework called KnowledgeSeeker. We have proposed and

implemented different ontological components and processes in the KnowledgeSeeker which are required to develop different kinds of ontology-based intelligent applications. The A* algorithm provides the accuracy and reduces the retrieval of rules or extraction of rules for similar web sites with minimal amount of time.

10. References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [1]. Where appropriate, include the name(s) of editors of referenced books.

- [1] "A New Method for Knowledge and Information Management Domain Ontology Graph Model" James N. K. Liu, Senior Member, IEEE, Yu-Lin He, Student Member, IEEE, Edward H. Y. Lim, and Xi-Zhao Wang, Senior Member, IEEE.
- [2] "A Method of Information Source Selection for Domain Ontology Construction" Jun Xing, Chuang Liu, Shiguo Xu, 2012 Third International Conference on Intelligent Control and Information Processing.
- [3] "Dealing With Uncertain Entities in Ontology Alignment Using Rough Sets" Sadaqat Jan, Member, IEEE, Maozhen Li, Hamed Al-Raweshidy, Senior Member, IEEE, Alireza Mousavi, Member, IEEE, and Man Qi " IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 42, NO. 6, NOVEMBER 2012"
- [4] "A Method of Information Source Selection for Domain Ontology Construction" Jun Xing, Chuang Liu, Shiguo Xu, 2012 Third International Conference on Intelligent Control and Information Processing.
- [5] "Ontology-based Reuse of Failure Modes for FMEA: Methodology and Tool", Guoqi Li School of Reliability and System Engineering Beihang University (BUAA) Beijing, China, 2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops.
- [6] "An Effective Repeated Rule Acquisition Using Rule Ontology from Similar Web Sites based on Genetic Algorithm", G.Senthil Kumar, K.K.Kanaga Mathan Mohan, International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 2, February 2013.