Fourth International Conference on Semantics, Knowledge and Grid

A Transport Service Ontology-based Focused Crawler

Hai Dong[1], Farookh Khadeer Hussain[2], Elizabeth Chang[3]

Digital Ecosystem and Business Intelligence Institute, Curtin University of Technology
GPO Box U1987, Perth, WA 6845, Australia
[1]hai.dong@cbs.curtin.edu.au
[2]farookh.hussain@cbs.curtin.edu.au
[3]elizabeth.chang@cbs.curtin.edu.au

Abstract— **Ontology is a technology for conceptualizing specific domain knowledge, which can provide machine-readable definitions to the severed domain. Therefore, ontology can be utilized to enhance the performance of focused crawlers, by precisely defining the crawling boundary. In this paper, we will exhibit a conceptual framework of an ontology-based focused crawler serving in the domain of transport services. Here, a transport service ontology is designed for filtering non-relevant metadata, by means of logically linking the metadata with ontological concepts. In addition, we will provide the evaluation process in order to assess the power of ontology in the focused crawler. Conclusion and further works based on our current evaluation results will be made in the final section.**

I. Introduction

A crawler is an agent which can automatically search and download webpages [3]. Focused (topical) crawlers are a group of distributed crawlers that specialize in certain specific topics. Each crawler will analyse its topical boundary when fetching webpages [1]. Ontology is a technology for conceptualizing specific domain knowledge, which can provide machine-readable definitions to the severed domain [5]. Thus, ontology can be utilized to precisely define the crawling boundary, which could enhance the performance of focused crawlers. In this paper, we will present a transport service ontology-based focused crawler.

The structure of the following context is organized as below:

First of all, we will review and analyse the past works in the field of new forms of World Wide Web integrated with semantic technologies, and ontology-based focused crawlers. Next, we will present the system architecture of the whole crawling system. Furthermore, we provide a brief introduction towards the transport service ontology and transport service metadata, followed by the illustration of an extended case-based reasoning algorithm for logically connecting metadata and ontological concepts. Subsequently, we will carry out the evaluation process to test the crawler's performance based on five selected indicators, followed by concluding the evaluation results. Finally, we will make a general conclusion to our research.

II. Related Works

In this section, we will briefly review the emerging forms of webs with semantic technologies, and ontology-based focused crawlers.

A. *Evolution of the World Wide Web*

After the World Wide Web emerged, researchers attempted to enhance its quality by various semantic technologies. Currently there are three new forms of recognized webs enhanced by various semantic technologies, which are semantic web, semantic grid, and knowledge grid.

Semantic web is "*a web of data*", which is used to express the meaning of web data by means of diverse ontological mark-up languages, such as XML, RDF, OWL and so forth. It provides the machine-understandable information for computers to retrieve, share and merge knowledge on the internet [18].

Semantic grid is the integration of semantic web, web service and grid technologies, in order to provide an internet-based platform for people and machines to organize, share, cluster, fuse and manage the distributed versatile semantic resources in the world. It is built upon the infrastructure of grid computing, by using semantic web technologies to enrich the semantic content of grid services, and web service technologies to enhance the function of grid service provision [12] [13].

Knowledge grid is an intelligent and sustainable internet-based environment that "*enables people and machines effectively capture, coordinate, publish, understand, share and manage knowledge resources*", by means of providing on-demand services for supporting scientific innovation, cooperative team work, problem solving and decision making [12]. Its core part is a Resource Space Model, which is a semantic model with the purpose of discovering and organizing knowledge resources by providing well-defined classification spaces to semantically cluster the retrieved knowledge resources [11] [13] [14] [15] [16] [17].

Currently all of these new forms of webs are still in research phase. The semantic grid and knowledge grid can be deemed as the extended versions of the semantic web, merged with other innovative technologies. In addition, since the knowledge grid and semantic grid have close goals, they can work in a collaborative environment, by making the

978-0-7695-3401-5/08 $25.00 © 2008 IEEE
DOI 10.1109/SKG.2008.56

IEEE computer society

knowledge grid work in the high level of the semantic grid [12].

B. Ontology-based focused crawlers

Generally speaking, ontology-based focused crawlers are a series of crawlers which utilize ontology to link the fetched web documents with the ontological concepts (topics), with the purpose of organizing and categorizing web documents, or filtering irrelevant webpages with regards to the topics. In the following we briefly review some focused crawlers that can be classified into the ontology-based ones.

A focused crawler – LSCrawler, was designed by Yuvarani et al., by using ontology to analyze the semantic similarity between URLs and topics. For each query, an ontology is retrieved for the crawling. Then the query is delivered to popular keyword search engines, e.g. Google and the retrieved URLs are sent back to the crawler. Based on the retrieved URLs, the webpages identified by the URLs are fetched and downloaded, and terms are abstracted from the webpages. Each set of terms extracted from a webpage are matched with the ontological concepts, in order to determine its relevance. Based on the relevance values, the URLs are ranked and then stored for further visit [9].

Tane et al. developed an ontology management system – Courseware Watchdog. One important component of the system is an ontology-based focused crawler. By means of the crawler, a user can specify his/her preference, by assigning weights to the concepts of an ontology. By means of the interrelations between concepts within the ontology, the weights of other concepts can be calculated. Once a webpage is fetched, its text and URL descriptions are matched with the weighted ontological concepts. Thus, the weights of the webpage and its URLs are measured, ranked and clustered according to the concepts. In addition, the webpage relations can be viewed by linking the webpages to the ontology concepts that appear in the webpages [7].

Ganesh proposes an association metric, with the purpose of optimizing the order of visited URLs for web crawlers. For each URL, an association metric evaluates its semantic content based on a reference domain-specific ontology. In addition, the metric of URL can analyze the link strength between parent and children webpage after the latter is downloaded, in order to refine itself [4].

THESUS aims to organize online documents by linking their URLs to hierarchical ontology concepts, which are seen as thematic subsets. A web crawler is used in the document acquisition component of the system. The mechanism of this crawler is as follows: first, the crawler extracts the URLs and their descriptive texts from the initial set of documents; then the descriptive text of one URL is matched with one of the ontological concepts, and the URL is linked to concept. A threshold of maximum times of recursions or maximum number of documents is set as an ending requirement [6].

Yang proposed a semantic web crawler program working in an ontology-based web environment. First of all, a knowledge base is designed, which stores ontologies. A web crawler then obtains all data from a given website. Next, the web information is modelled, which contains a website profile and all associated webpage profiles. Each profile includes the basic description, static information, and ontological information regarding a corresponding webpage. To realize this objective, a DocExtractor program is designed to extract the basic information from a webpage for the first section, calculating statistical data for the second section and remove all HTML tags. Subsequently, an OntoAnnotator is used to annotate the web metadata for the third section. Within the DocExtractor, a HTML Analyzer is utilized to analyze the webpages from a DocPool which contains the webpages from the retrieved website, then extracts the information regarding URLs, titles, anchors and headings, and calculates the statistical data regarding tags. Thereafter a HTML Tag Filter is used to remove all tags from the analyzed webpages, and a Document Parser converts the tag-free webpages into a list of keywords. These keywords are passed to an OntoAnnotator. In the OntoAnnotator, an OntoClassifier is used to describe each webpage with the mostly matched classes of domain ontology based on the tf-idf algorithm. Finally, an Annotator is used to annotate the webpage with the classes and their frequencies, and a Domain Marker is used to determine the belonged domain, based on the class frequencies for the webpage [8].

From the above cases, it is observed that most of the crawlers are designed for general domain. Some of them are encapsulated in larger systems, and others are designed as separate tools. Ontology is mostly used to match the fetched URLs or webpages with the predefined topics (ontological concepts), by means of computing the similarity values between the ontological concepts and the fetched URLs or webpages.

One common issue is that most of these crawlers (3 of 5) do not provide evaluation details, which cannot prove their superior performance, compared with traditional focused crawlers. With regard to evaluation methods, Yuvarani's experiment provides us with an impressive idea, namely adopting traditional information retrieval evaluation benchmarks, such as harvest rate, precision, and so forth.

Thus, to solve the above issue, apart from presenting an ontology-based semantic crawler serving in the domain of transport service, we also provide benchmarks for evaluating purpose.

III. System Architecture

The ontology-based focused crawler has three main parts – a focused crawler, a transport service ontology base, and a transport service metadata base (Fig. 1). The first part is to abstract the transport service metadata and cluster the metadata according to the transport service ontology. The second and third parts are used for storing the ontology and metadata respectively.

The workflow of the whole system is as below:

Step 1: The focused crawler downloads all webpages from a website in the internet.

Step 2: The focused crawler extracts all required information from the downloaded webpages, and yield transport service metadata based on the information. The metadata are then stored in the metadata base.

Step 3: The focused crawler computes the similarity values between each concrete ontological concepts in the ontology base and each metadata. If a similarity value is above a threshold, the corresponding metadata and ontological concept will be determined as logically related.

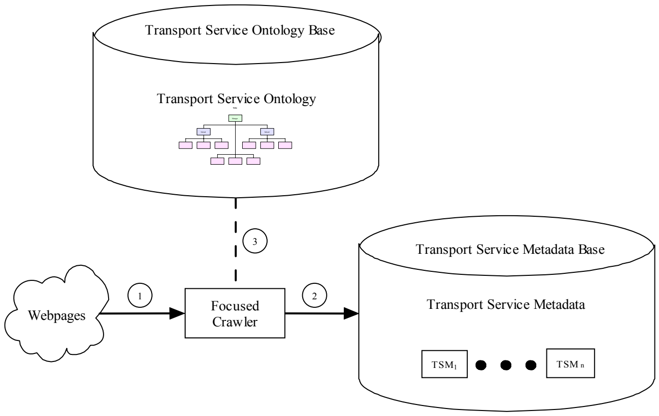
Transport Service Ontology Base
Transport Service Ontology
3
Transport Service Metadata Base
Webpages
1
Focused Crawler
2
Transport Service Metadata
TSM$_1$
TSM $_n$

Fig. 1. System architecture

In the following sections, we will reveal the mechanism of the transport service ontology base, transport service metadata base and focused crawler in detail.

IV. Transport Service Metadata and Transport Service Ontology

In this section, we will discuss the technical details regarding transport service metadata and transport service ontology.

A. Transport service metadata

The major mission of transport service metadata is to draw the meaningful information regarding transport service from downloaded webpages. Fig. 2 illustrates the format of the transport service metadata, where transport service metadata can be represented as a tuple where the elements of the tuple can be complex elements as defined below:

[Linked Concepts, Service Provider Name, Provider Address, Provider Contact Detail, Service Description] where

Linked Concepts refer to the references to the semantically linked concepts.

Service Provider Name refers to the name of the person or organization that provides a service.

Provider Address refers to the address where a service provider can be located.

Provider Contact Details refer to the information regarding how a service provider is contacted, for instance, mail box, phone number, fax number, website and so on.

Service Description refers to the detailed text description with regards to the content of a service. This can be used for matching with a service concept, which will be described later.

Transport Service Metadata
Linked Concepts Service Provider Name Provider Address Provider Contact Details Service Description

Fig. 2. Transport service metadata format

B. Transport service ontology

The primary mission of the transport service ontology is to cluster the conducted metadata by building logical links to its concepts. The structure of transport service ontology is a four-layer hierarchy (Fig. 3). The first layer is the root of the hierarchy, which represents the abstract concept of all services in the transport domain. The second layer is the preliminary specialization for the abstract transport service concept, which classifies four categories of transport service concept – air transport, rail transport, road transport, and shipping services. These can be also regarded as the four basic sub-domains of transport services. The service concepts in this layer are still considered as abstract. The third layer is the further specialization for the abstract transport service concepts in the second layer, which represents the services in each basic sub-domain of transport services. In this layer, the service concepts could be regarded as concrete or abstract concepts. For each concrete service concept, it corresponds to the actual services in the real social environment (e.g., in Fig. 3, aircraft charter/rental service); for each abstract service concept, it has further specializations which are all concrete services in the fourth layer (e.g., in Fig. 3, air cargo service-abstract has air cargo custom clearance service, air cargo service and so forth as its specializations).

In conclusion, the service concepts in the first and second layer are all abstract concepts, which execute the function of service domains definition, the service concepts in the third layer are hybrid concepts between concrete concepts and abstract concepts, which are used to restrict service domain and correspond to the actual services in the real world; the service concepts are concrete concepts in the bottom layer. Each concrete service concepts has the attribute of concept description, which refers to the detailed description of the corresponding service. This attribute can be used to for matching with the semantically similar transport service metadata, which will be discussed later.

V. Extended Case-Based Reasoning Algorithm

As described earlier, one mission of the focused crawler is to compute the similarity values between each concrete ontological concepts and metadata, by comparing the attribute of concept descriptions from concepts and the attribute of service descriptions from metadata. The similarity value computing is based on an Extended Case-Based Reasoning (ECBR) Algorithm.

CBR model is used to retrieve and reuse the existing problem solutions for emerging problems, which has four sub-processes as below [2]:

Retrieve: a new problem is matched with cases in database.

Reuse: if there are cases matched, the solutions to the retrieved cases are reused as the solutions of the emerging problem.

Revise: if the retrieved cases cannot completely match the problems, the solutions to the problem need to be revised.

Retain: the new case, incorporating with both problems and solutions, is stored in database.

Every feature extracted from incident reports is awarded an equal weight. Every feature in a new incident is compared with the corresponding feature in each of the other incidents. If the features match, a score of 1 is awarded. If the features do not match, a score of 0 is awarded. A similarity score is calculated by:

1. Finding the sum of the matching features;
2. Dividing this sum by the number of features contained in the incident, as in the formula below:

$$sim(T,S) = \frac{\sum_{i=1}^{n} f_i(T_i, S_i)}{n}$$

Then a threshold is set up to determine whether the two incidents are matched or not.

Here we extend the use of CBR model to the field of information retrieval. It is well-known that most information retrieval models are built on the basis of index terms. Thus, to use the ECBR model for the metadata-concept matching, an index term list needs to be generated. The following is the definition of an ECBR model.

First of all, a list of index terms are generated from all concept descriptions in the transport service ontology. Every index term in a concept description and service description is awarded as an equal weight. Every index term of a service description are matched with every index term of a concept description. If one couple of index terms are matched, a score of 1 is awarded; otherwise a score of 0 is awarded. A similarity score is calculated by summing the scores and then being normalized by the number of index terms appeared in the concept descriptions, as in the formula below.

The similarity value of a concept description c_j to a service description s is calculated as

$$sim(c_j, s) = \frac{\sum_{j=1}^{m} f(c_{k_j}, s)}{m}$$

$$f(c_{k_j}, s) = \begin{cases} 1 \ if \ \exists s_k | (\forall k_i, g_i(c_{k_j}) = g_i(s_k)) \\ 0 \ otherwise \end{cases}$$

where c_j is the content of the concept description attribute regarding a transport service concept, k_i is an index term, $c_j = (c_{k_1}, c_{k_2} ... c_{k_m})$, c_k is the index terms of c_j, m is the number of index terms of c_j. s is the content of the service description attribute regarding a transport service metadata, $s = (s_{k_1}, s_{k_2} ... s_{k_n})$, s_k is the index terms of s, n is the number of index terms of s, g_i is a function that returns weight associated with k_i.

VI. Prototype Implementation

The prototype implementation phase contains two sub-processes, which are transport service ontology building and focused crawler implementation.

The first process is realized by using Protégé-owl, which can be seen in Fig. 4.

The second process is realized by using Java language. After all webpages from a website are downloaded, the focused crawler will parse the webpage texts by analyzing HTML, XHTML tags and keywords in the texts. Then the crawler will extract the meaningful information and transform it to metadata by adding OWL tags. Next, the ECBR algorithm will be used to compute the similarity values between each concept and metadata. If a similarity value is above a threshold value, the URI of the concept will be stored

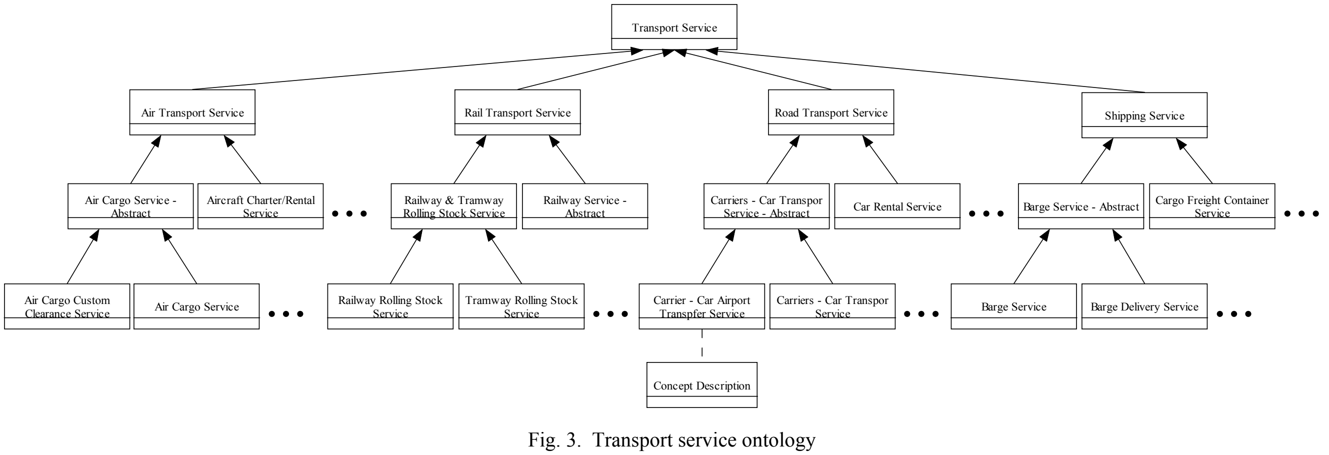
Transport Service
Air Transport Service
Rail Transport Service
Road Transport Service
Shipping Service
Air Cargo Service - Abstract
Aircraft Charter/Rental Service
Railway & Tramway Rolling Stock Service
Railway Service - Abstract
Carriers - Car Transpor Service - Abstract
Car Rental Service
Barge Service - Abstract
Cargo Freight Container Service
Air Cargo Custom Clearance Service
Air Cargo Service
Railway Rolling Stock Service
Tramway Rolling Stock Service
Carrier - Car Airport Transpfer Service
Carriers - Car Transpor Service
Barge Service
Barge Delivery Service
Concept Description

Fig. 3. Transport service ontology

into the attribute of linked concepts of the corresponding metadata.

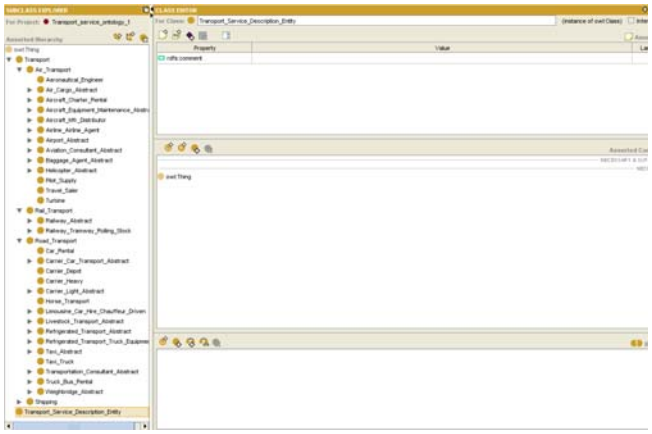

Fig. 4. Screenshot of transport service ontology in Protégé-owl

VII. Evaluation

To evaluate the performance of our focused crawler, we select five performance indicators from the traditional information retrieval evaluation approaches and then perform a series of experiments based on them.

A. Performance indicators

To evaluate our ontology-based focused crawler, five performance indicators of information retrieval are used, which are harvest rate, precision, recall, mean average precision and fallout rate.

Harvest rate in the information retrieval is used to measure the crawling ability of a crawler in a collection. In this case, harvest rate is the proportion of logically linked metadata in the whole collection of metadata, which can be represented below:

$$\text{Harvest rate} = \frac{\text{number of logically linked metadata}}{\text{number of metadata}}$$

Precision in the information retrieval is used to measure the preciseness of a retrieval system [10]. Precision for a single concept is the proportion of logically linked, at the same time, and relevant metadata in all logically linked metadata to the concept, which can be represented below:

$$\text{Precision(S)} = \frac{\text{number of logically linked relevant metadata}}{\text{number of logically linked metadata}}$$

With regard to the whole collection of concepts, the whole precision is the sum of precision for each concept normalized by the number of concepts in the collection, which can be represented below:

$$\text{Precision(W)}=\frac{\sum_{i=1}^{n}\text{Precision}(\text{S}_i)}{n}$$

Recall in the information retrieval refers to the measure of effectiveness of a query system [10]. Recall for a single concept is the proportion of logically linked, and at the same time, relevant metadata in all relevant metadata, which can be represented below:

$$\text{Recall(S)} = \frac{\text{number of logically linked relevant metadata}}{\text{number of relevant metadata}}$$

With regard to the whole collection of concepts, the whole recall is the sum of recall for each concept normalized by the number of concepts in the collection, which can be represented below:

$$\text{Recall(W)}=\frac{\sum_{i=1}^{n}\text{Recall}(\text{S}_i)}{n}$$

Before we introduce the definition of mean average precision, the concept of average precision should be defined. Average precision for a single concept is the average of precision values at each logically linked and relevant metadata for this concept, given that these metadata are ranked according to their computed similarity values. This indicator is used to measure how quick and precise a crawler works [10], which can be represented below:

$$\text{average precision(S)} \\ = \frac{\text{sum(precision @ logically linked relevant metadata)}}{\text{no. of logically linked relevant metadta}}$$

Mean average precision refers to the average of average precision values for the whole collection of concepts, which can be represented below:

$$\text{Mean average precision}=\frac{\sum_{i=1}^{n}\text{Average precision}(\text{S}_i)}{n}$$

All of the above indicators have the same limitation – they do not consider the number of non-relevant metadata in a retrieved collection. In addition, if there is no relevant metadata in the retrieved collection, recall cannot be defined. To solve this issue, we need another performance indicator – fallout rate. The fallout rate for a single concept which is proportion of logically linked, at the same time, and non-relevant metadata in the whole collection of non-relevant metadata to the concept, which is represented below:

$$\text{Fallout rate(S)} = \frac{\text{No. of logically linked non-relevant metadata}}{\text{No. of non-relevant metadata}}$$

With regard to the whole collection of concepts, the whole precision is the sum of fallout rate for each concept normalized by the number of concepts in the collection, which can be represented as:

$$\text{Fallout rate(W)}=\frac{\sum_{i=1}^{n}\text{Fallout rate}(S_i)}{n}$$

Opposite to other performance indicators, the lower the fall rate value is, the better the crawler's performance has.

The following experiment will be executed based on the five performance indicators.

B. Experiment

As mentioned earlier, after the similarity values between each concept and each metadata have been obtained, a threshold is set up to determine whether the concept and metadata should be logically linked or not. To obtain the most proper threshold values, our evaluation concentrates on testing the five performance indicators along with different threshold values.

To test the performance of the ontology-based focused crawler, we choose the Australian Yellow Page® website as the testing resource, and we use our crawler to download 1000 company webpages under the category of transport in the website.

The results of indicators are shown from Fig. 5 to Fig. 9.

The focused crawler totally abstracts 1982 metadata from the 1000 business webpages. The figure of harvest rate is shown in Fig. 5. It is observed that the harvest rate slightly drops from 98.44% to 94.40%, when the threshold values rises from 0.5 to 1. In addition, the curve stably remains at the bottom after the threshold value rises from 0.7. The reason behind this phenomenon is that, along with the rise of threshold, more noise metadata (metadata that are relevant to any ontological concepts) have been filtered and thus reduce the harvest rate. It is found that the threshold value does not heavily impact the harvest rate. One of the reasons is that there are not too many noise metadata in our testing data. Since the harvest rate values are all above 90% and keep relatively steady, it is not regarded as an important factor to determine the threshold.

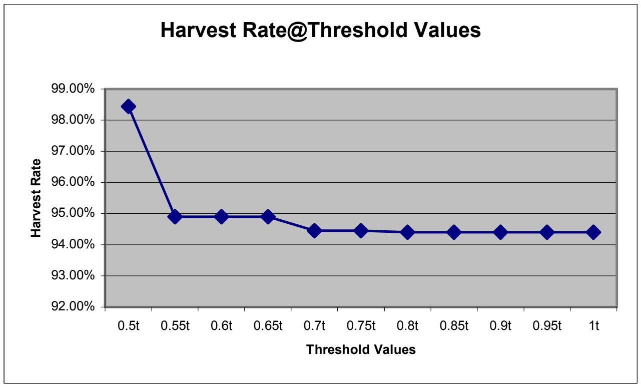

Fig. 5. Harvest rate @ threshold values

The harvest rate proves that the focused crawler presents a prime performance in linking the metadata with the ontological concepts. However, besides the test of the crawler's capacity, we also need to test its effectiveness. Precision provides us with an opportunity to evaluate the crawler from another perspective. It is observed that, form Fig. 6, the precision critically relies on the threshold value – the precision quickly jumps from 14.62% to 89.53% when the threshold value rises from 0.5 to 1. Similarly, the precision reaches to the top point and keeps steady after the threshold value reaches to 0.8.

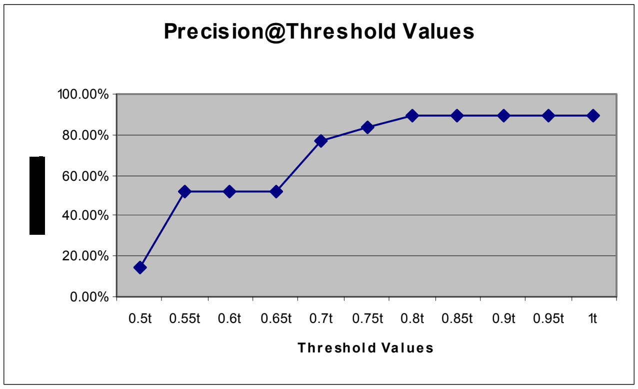

Fig. 6. Precision @ threshold values

Differing from precision, recall is to test the crawling effectiveness of a crawler. It is observed that, from Fig. 7, the recall value drops down slightly from 98.61% to 95.78, due to the rise of the threshold value. This is because high threshold values may block the logical linkage between concepts and relevant metadata, due to their relatively low similarity values computed by the ECBR algorithm. However, this impact is relatively weak compared with the indicator of precision, as the recall only has varied 2.83% in the process of recall variation. In addition, as a whole, the recall level is above 90%, which is an outstanding performance. Thus, same as harvest rate, recall is not a predominant factor to determine the threshold value.

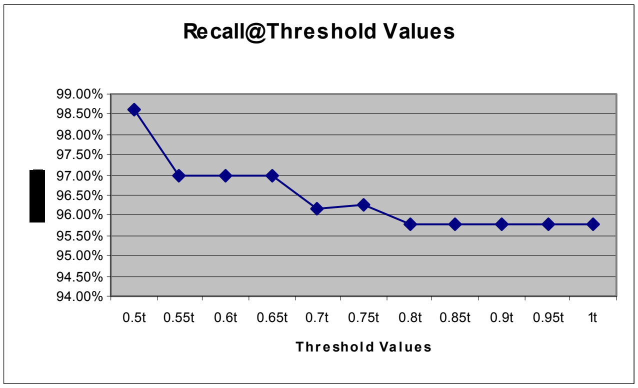

Fig. 7. Recall @ threshold values

Fig. 8 displays the variational trend of the mean average precision, which looks similar to the trend precision in Fig. 6, even though they are measured for different purposes. The difference is that the variational range of mean average precision is smaller than the precision, which is from 18.77% to 86.17%. Similarly, when the threshold value reaches 0.8, the curve of mean average precision remains steady on top. It is revealed that the threshold has a similar positive impact on mean average precision as it has on precision.

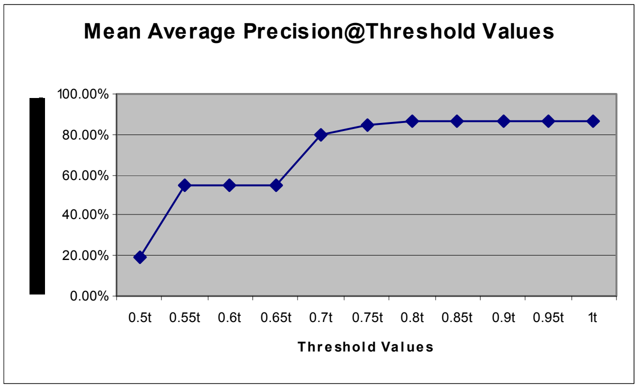

Fig. 8. Mean average precision @ threshold values

By observing the curve of fallout rate @ threshold values in Fig. 9, we find that the growth of threshold value obviously cuts down the fallout rate, which declines from 5.08% to 0.07%. When the threshold is above 0.75, the fallout rate value closes to 0, which is a superior performance for the crawler.

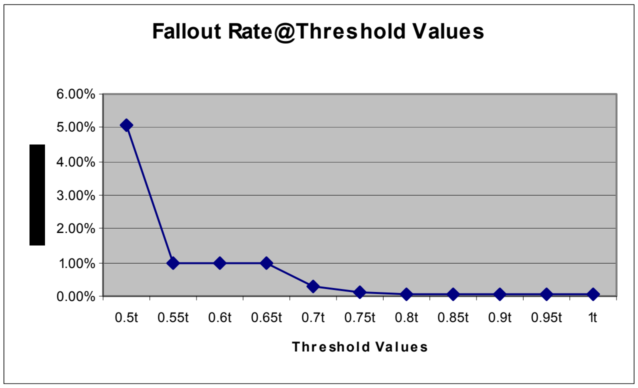

Fig. 9. Fallout rate @ threshold values

By means of the five performance indicators, it is discovered that the point that threshold value is equal to 0.8 is an obvious boundary for the crawler's overall performance. At this point, the precision and mean average precision curves remains stable on top, and the harvest rate, recall, and fallout rate values steadily remain at their lowest values. As discussed before, owing to the tiny changes of harvest rate and recall, they do not need to be considered when determining the threshold value. The precision, mean average precision and fallout rate are in their top performance in this point, which indicates that 0.8 is a relatively proper threshold value for the focused crawler. As a conclusion, in this experiment, the crawler shows convincing results in all five performance indicators, as the peak values of precision and mean average precision reach 90%, the overall values of harvest rate and recall remain stable above 90%, and the fallout rate close to 0 in its lowest point.

VIII. Conclusions and Further Works

In this paper, we deliver a conceptual model of an ontology-based focused crawler serving in the domain of transport services. By reviewing past works in this area, we find that ontology can be used to make efforts to restrict the crawling scope. However, most researches do not provide detailed evaluation results, which cannot completely reveal the power of ontology. Thus, to solve this problem, apart from presenting the conceptual framework, we also provide the indicators for evaluating the crawler. The whole crawling system consists of three parts – a focused crawler, a transport service ontology base, and a transport service metadata base. The focused crawler takes the responsibility of downloading webpages, analyzing and parsing web documents, extracting meaningful information from the documents and forming metadata based on the information, and logically linking the metadata and ontological concepts; the ontology base is for storing a transport service ontology, with the purpose of limiting the crawling scope; the metadata base is for storing the transport service metadata and the links of metadata to concepts. The ECBR algorithm for the logical links is derived from the CBR algorithm in the datamining field, by rebuilding it on the basis of index terms. It is found that the existing assessment methods for ontology-based crawlers are based on the traditional information retrieval evaluation methods. Therefore, we choose five performance indicators from those methods for the crawler's evaluation – harvest rate, precision, recall, average precision, and fallout rate. The first indicator is to test the focused crawler's crawling capacity; the second indicator is to test the crawler's crawling efficiency; the third indicator is to test the crawler's crawling effectiveness; the fourth indicator is to synchronously test the crawler's crawling quickness and efficiency; the fifth indicator is to make up the common limitation of the prior four indicators, which is to consider the proportion of non-relevant metadata in the whole collection. The five indicators are used to find out the most proper threshold for determining whether a couple of metadata and concept should be logically linked or not. We use the crawler to crawl 1000 business webpages under the transport category of the Australian Yellow Page® website. The results show that the crawler performs well in all of the five performance indicators when the threshold is set to 0.8. Therefore, in this experiment, we directly prove the significant effort of ontology and ECBR algorithm in focused crawlers.

One limitations of our research is that the volume of the testing webpages is not very big. This issue will be solved in the forthcoming experiments. In addition, similar to other classic index term-based algorithms, ECBR algorithm also relies on the independency of index terms, which cannot be perfectly realized in the real environment. However, the current testing proves that the index term independency has not critically impacted the crawler's performance. Thus, the impact of index term independency needs to be researched further.

Acknowledgment

We would like to express our gratitude to the assistance of all relevant DEBII staff, especially to our programmer Wei Liu who took the responsibility of implementing the focused crawler's prototype and testing benchmarks.

References

[1] A. A. Barfourosh, M. L. Anderson, H. R. M. Nezhad, and D. Perlis, "Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition," Department of Computer Science, University of Maryland, Maryland 2002.

[2] D. C. J. Carthy, A. Drummond, J. Dunnion, and J. Sheppard, "The use of data mining in the design and implementation of an incident report retrieval system," in *Systems and Information Engineering Design Symposium*, Charlottesville, 2003, pp. 13-18.

[3] J. Cho and H. Garcia-Molina, "Parallel Crawlers," in *WWW2002*, Honolulu, 2002, pp. 124-135.

[4] S. Ganesh, M. Jayaraj, V. Kalyan, and G. Aghila, "Ontology-based web crawler," in *the International Conference on Information Technology: Coding and Computing (ITCC'04)*, Las Vegas, 2004.

[5] T. Gruber, "Ontology," in *Encyclopedia of Database Systems*: Springer-Verlag, 2008.

[6] M. Halkidi, B. Nguyen, I. Varlamis, and M. Vazirgiannis, "THESUS: organizing web document collections based on link semantics," *The VLDB Journal,* vol. 12, pp. 320–332, 2003.

[7] J. Tane, C. Schmitz, and G. Stumme, "Semantic resource management for the web: an elearning application," in *WWW2004*, New York, 2004.

[8] S.-Y. Yang, "An ontological website models-supported search agent for web services," *Expert Systems with Applications*, vol. In Press, Corrected Proof.

[9] M. Yuvarani, N. C. S. N. Iyengar, and A. Kannan, "LSCrawler: a framework for an enhanced focused web crawler based on link semantics," in *the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)*, 2006.

[10] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: Addison-Wesley, 1999.

[11] H. Zhuge, "Resource space grid: model, method and platform," *Concurrency and Computation: Practice and Experience,* vol. 16, pp. 1385-1413, 2004.

[12] H. Zhuge, *The Knowledge Grid*. Singapore: World Scientific, 2004.

[13] H. Zhuge, "Semantic grid: scientific issues, infrastructure, and methodology," *Communications of the ACM,* vol. 48, pp. 117-119, 2005.

[14] H. Zhuge, "Transformation from OWL to RSM," in *1st Asia Semantic Web Conference (ASWC2006)*, Beijing, 2007, pp. 4-23.

[15] H. Zhuge, P. Shi, Y. Xing, and C. He, *The Web Resource Space Model*: Springer-Verlag, 2007.

[16] H. Zhuge, Y. Xing, and P. Shi, "Resource space model, OWL and database: mapping and integration," ACM Transactions on Internet Technology, vol. 8, 2008.

[17] H. Zhuge, E. Yao, Y. Xing, and J. Liu, "Extended resource space model.," *Future Generation Computer Systems,* vol. 21, pp. 189-198, 2005.

[18] T. Berners-Lee, D. Connolly, S. Hawke, I. Herman, E. Prud'hommeaux, and R. Swick, "W3C semantic web Activity," W3C, 2008.