An Ontology-based Webpage Classification Approach for the Knowledge Grid Environment

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

Digital Ecosystems 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*— With the rapid growth of the amount of information available in the Web, webpage classification technologies are widely employed by many search engines in order to formulate user queries and make users' search tasks easier. Knowledge Grid is a new form of Web environment, in which a Resource Space Model is employed in order to classify available semantic documents within the Web environment. However, it is well known that the semantic documents are proportionally small in relation to the whole Web documents, and the Resource Space Model cannot process these Web documents without semantic supports. In order to solve the above issue, in this paper, we present a novel ontology-based webpage classification method for the Knowledge Grid environment, which utilizes generated metadata from webpages as the intermedium to classify the webpages by ontology concepts. We design a conceptual model of a Webpage Classification Agent and build the prototype in a chosen domain. A series of experiments have been conducted using the prototype in order to evaluate the conceptual model. Conclusions about the evaluation are drawn in the final section.**

I. INTRODUCTION

With the rapid increase in the number of Web users and internet service providers (ISPs), the information available in the Web appears to have reached an astronomical number (2.93×10^{21} bits in January 2009 - 10 times bigger than for 2006 according to IDC's report [1]). Numerous Web search engines have been developed in order to assist users to explore Web resources. Moreover, classification/categorization techniques are widely utilized in these search engines in order to assess the themes of websites and facilitate the search task. Web directories such as Yahoo! and Yellowpages® are representative examples of classification/categorization in the Web. Traditionally, the classification/categorization tasks are manually performed by domain experts. Nevertheless, it is impossible to keep the tasks within the realm of execution by a human, given the rapid growth of Web information. Therefore, as the Web keeps growing, classification/categorization becomes increasingly important and its automation has already been regarded as a necessary undertaking.

While it may not be currently feasible to extract the full meaning of a webpage, intelligent software agents have been developed to extract the features of a webpage and employ them to classify and categorize the webpages [2]. The webpage classification/categorization can then be employed to formulate queries, to organize bookmark files, or to construct user profiles.

Knowledge Grid is a new form of Web, which is an intelligent and sustainable internet-based environment that "*enables people and machines to effectively capture, coordinate, publish, understand, share and manage knowledge resources*", by providing on-demand services for supporting scientific innovation, cooperative team work, problem solving and decision making [3]. At its core is a Resource Space Model (RSM), which is a semantic model, the purpose of which is to discover and organize knowledge resources by providing well-defined classification spaces to semantically classify the retrieved knowledge resources [3-8]. However, one limitation of the RSM needs to be addressed as follows.

The input of the RSM are Web Ontology Language (OWL) descriptions, which are OWL annotated resources (text, images, videos, etc.) [7]. However, due to a lack of technologies to extract and annotate the semantics from normal Web resources, most of which are Hyper Text Mark-up Language (HTML) annotated Webpages, the RSM does not have the ability to classify most of the available Web documents. As a result, numerous Web documents without semantic supports are ignored by the RSM.

Therefore, in this paper, in order to solve the above issue, we present a novel ontology-based webpage classification approach for the Knowledge Grid environment. This approach is able to collect OWL descriptions for semantic-less webpages, which provides a broader scope for the input of the RSM.

The rest of the paper is organized as follows: in Section 2 we will review several forms of Web and the existing webpage classification technologies, as well as analyse the issues within them; in Section 3, we will deliver a two-step webpage classification method and the system architecture of a Webpage Classification Agent; in Section 4, we will introduce a unified metadata format and an ontology concept format; in Section 5, we will provide an information retrieval algorithm for matching between webpages and similar ontology concepts; in Section 6, we will implement the prototype in a chosen domain and make evaluations based on the prototype; conclusion are drawn and future work is suggested in Section 7.

978-0-7695-3810-5/09 $26.00 © 2009 IEEE
DOI 10.1109/SKG.2009.69

II. Related Works

Webpage classification/categorization refers to the process of assigning a webpage to one or more predefined category labels [9]. In this section, we will briefly review and analyse the existing webpage classification technologies.

The existing webpage classification research emphasizes two main perspectives:

(1) one adopts traditional document classification models for webpage classification;

(2) another perspective studies webpage classification based on the features of webpages.

Existing document classification techniques can be mainly classified into two categories as follows.

The first classifies documents by providing additional information to them, which contains the following three subcategories:

(1) Web documents that can be indexed according to traditional index term-based methods, such as algebraic models, Bayesian models, which can be further referenced from [10-12];

(2) electronic documents that can be categorized according to their predefined attributes, e.g., title, subject, author, etc.;

(3) webpages that can use user-specified attributes to show their relevance to items, such as query history, user profile, etc [13].

The second is to directly measure the similarity between documents, which can be realized by transforming a multi-dimensional document into a 2-D or 3-D space by means of aggregating similar documents under the same themes. Chen et al. [14] present a typical example of the classification method. They designed a prototype of MetadataSpider, which utilizes an Arizona Noun Parser for extracting nouns and a SOM algorithm for classification webpages into different regions on a 2-D map.

The limitation of the traditional document classification methodologies to webpage classification can be concluded as follows.

First, they ignore the structure of webpages within websites, which may affect the similarity measure of webpages. In a website, similar webpages could be linked by hyperlinks, which can be considered as an important factor for webpage classification.

Second, the inter-document similarity measures are time-costing. Provided that some Web documents have high volume of contents, the inter-document comparisons often needs higher computing cost.

Given the two limitations above, many webpage classification methodologies based on website structure emerge, which analyse the paths of hyperlinks between webpages [15, 16]. Current researches focus on measuring the distance between nodes (webpages) in a graph (website). These distance-based methods primarily employ the means of k-means analysis, hierarchical classification [17] and k-nearest neighbour classification (k-NN) [18].

Kwon and Lee [19] provide a typical example of the k-NN-based website classification model, which involves three sequential steps as follows:

Step 1. Webpage selection. Two sub-steps are contained in this step: 1) the boundary of a website is detected by two author-defined restriction rules; 2) the most representative webpages are selected by assessing and ranking the connectivity of each webpage within the website boundary.

Step 2. Webpage classification. Two sub-steps are involved in this step: 1) the similarities of neighbour documents to a training document are calculated by the k-NN, and the similarities can be regarded as the similarities of the document to the categories pre-assigned to each neighbour document; 2) the likelihood of each category is estimated by summing up the k-nearest documents.

Step 3. Website classification. A website can be classified by multiple categories by summing up the similarities of its inner webpages for each category.

There are two issues for the distance-based method as pointed out by Boley et al. [2] as follows:

(1) These methods are significant in defining a distance measure in a multi-dimensional space. The distance-based methods deal only with the classification in a 2-D space, which may meet with difficulties when dealing with the classification in a multi-dimensional space.

(2) Large sizes of documents can produce large sizes of terms, which is costly in document similarity computing. The k-NN method still partly relies on the similarity measure between two documents, and thus inherits its defect.

III. System Architecture

In the previous section, we reviewed the existing researches in the field of webpage classification and discovered some issues arising from them. In this section, we will present a novel ontology-based webpage classification approach.

A. A Two-Step Webpage Classification Process

The webpage classification process involves two main steps as indicated in Fig. 1, which are described as follows:

Step 1. Metadata generation. Before the methodology is applied, webpages in the Web are linked by the Uniform Resource Locators (URLs). However, the URLs have a lack of semantics, so cannot be used to interpret the semantics and knowledge structure between webpages. In order to interpret the semantic relationships between webpages, first of all, we need to obtain the key feature information of the webpages. Thus, in this step, we define a unified metadata format for a specific domain, in terms of the OWL. By means of this metadata format, we may extract the feature information from each webpage and generate metadata based on the feature information. Then these metadata are stored in a Metadata Base. It needs to be noted that each metadata has a relationship with the webpage from which it has been extracted, and each webpage may sometimes have more than one related metadata. By means of the metadata generation,

each webpage is assigned some degree of semantics by its metadata. However, these webpages still lack the support of semantic relationships.

Step 2. Semantic links. In this step, we employ the predefined ontologies stored in a Knowledge Base, in order to enhance the semantics of the webpage structure. The ontology can provide domain knowledge to webpages by semantically linking them to their metadata. The semantic link is realized by exchanging the Unfirm Resource Identifiers (URIs) of mutually matched ontology concepts and metadata. The mutual matching is based on computing the similarities between ontology concepts and metadata by means of an information retrieval algorithm. Here we employ an Extended Case-based Reasoning (ECBR) algorithm, which will be introduced in Section 5.

Therefore, by means of the two steps above, webpages in the Web can be classified by the domain ontologies, which utilizes generated metadata as an intermedium between them.

B. System Architecture

In order to realize the two steps above, we design the conceptual framework of a Webpage Classification Agent, which is shown in Fig. 2. The agent's framework consists of four main parts as follows:

(1) Webpage Fetcher. The mission of the Webpage Fetcher is to download Web documents according to predefined URL lists.

(2) Webpage Parser. The Webpage Parser is designed to parse the downloaded Web documents into information pieces according to the predefined Web document parsing rules. The Web document parsing rules are defined by analysing relevant HTML tags in the Web documents.

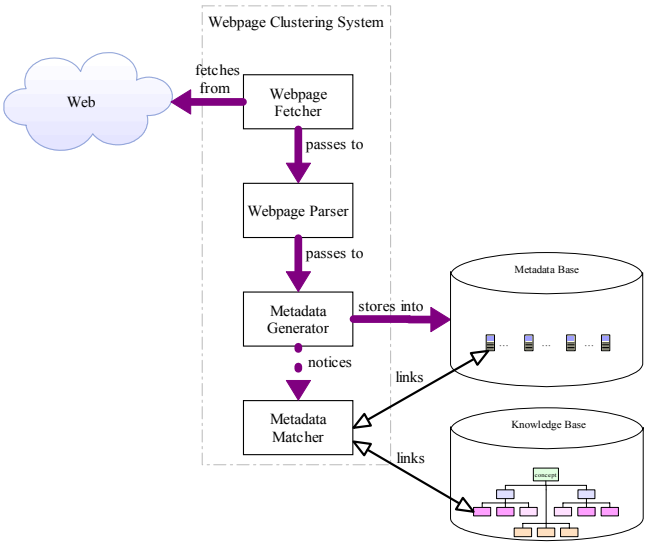

Fig. 2. System architecture of the Webpage Classification Agent

(3) Metadata Generator. The Metadata Generator is used to generate metadata by annotating the information pieces with OWL tags. The annotation is implemented based on the predefined metadata format.

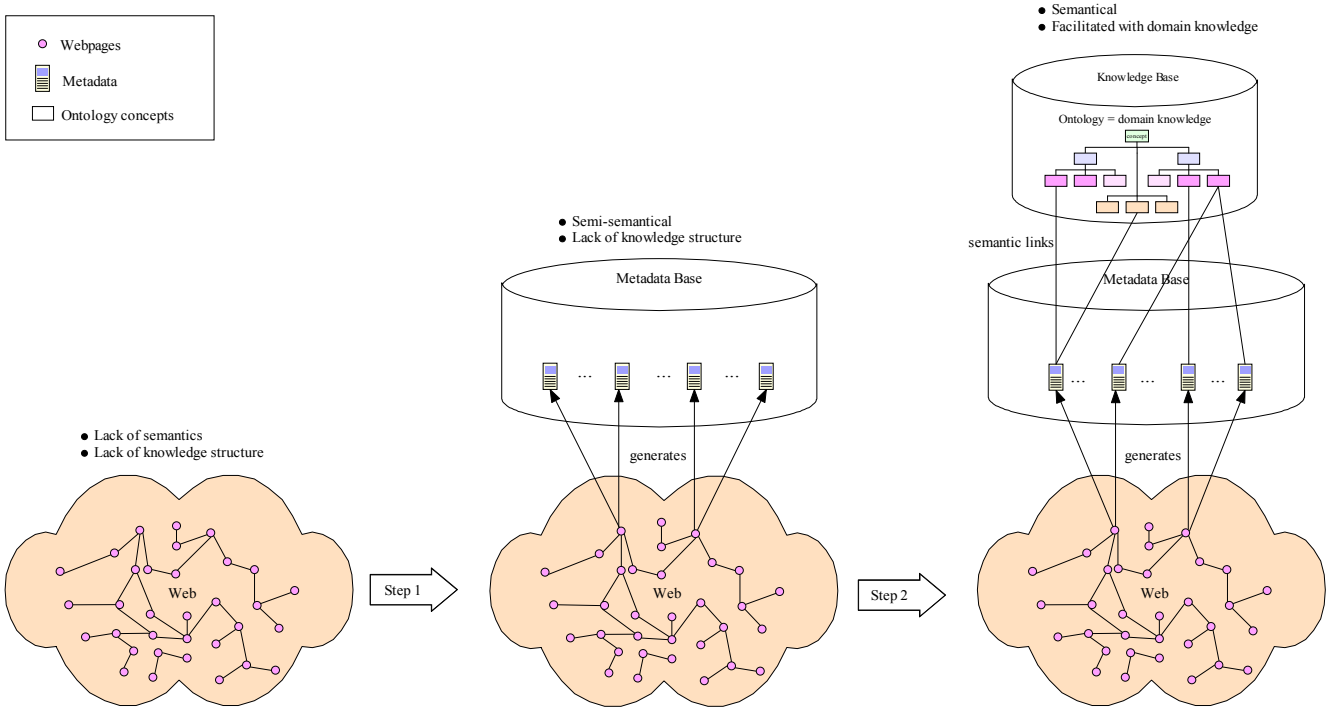

Fig. 1. Example of the two-step webpage classification process

(4) Metadata Matcher. Metadata Matcher is deployed to link service concepts with similar service metadata. The matching is based on the similarities between each concept and metadata computed by an information retrieval algorithm. By means of exchanging each of their URIs, the metadata and concepts can be linked.

From Fig. 2, we can observe that Parts 1-3 finish Step 1 in Section 3 namely metadata generation, and Part 4 completes Step 2, namely semantic links.

IV. Metadata and Ontology Concept Format

As mentioned in Section 3, a unified metadata format needs to be defined for metadata generation. As a matter of fact, there are many metadata formats available for webpage annotation, such as Resource Description Framework - in - attributes (RDFa) and Gleaning Resource Descriptions from Dialects of Languages (GRDDL). There is a limitation of the metadata formats as follows:

RDFa and GRDDL are all designed for annotating (Extensible Hypertext Mark-up Language) XHTML documents, which are XML-annotated HTML documents. However, in the web there are a large proportion of web documents which are not XHTML-annotated. Thus, these metadata formats cannot deal with the pure HTML documents.

Therefore, we need to design such a unified metadata format to cope with general web documents in the web. Similarly, we also need to define a unified ontology concept format for computing the similarity between metadata and concepts.

A. Metadata Format

Each metadata has two primary properties, which are metadataDescription and linkedConcepts.

metadataDescription is a data type property of metadata, which refers to the description of a metadata. The content of this property is formed by the Metadata Generator, by extracting meaningful information from webpages. Similar to its concepts counterpart, this property is also used to compute similarity values between metadata and concepts.

linkedConcepts is an object property of metadata, which is used to store the URIs of linked concepts. This property is the inverse of the linkedMetadata property in concepts. In other words, if a metadata stores a concept's URI in the linkedConcepts property, the concept must automatically have the metadata's URI in its linkedMetadata property.

The OWL code of metadata format is shown below:

```
<owl:Class rdf:ID="Metadata"/>
  <owl:DatatypeProperty
rdf:ID="metadataDescription">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#st
ring"/>
    <rdfs:domain rdf:resource="#Metadata"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:about="#linkedConcepts">
    <owl:inverseOf rdf:resource="#linkedMetadata"/>
    <rdfs:domain rdf:resource="#Metadata"/>
    <rdfs:range rdf:resource="#Concept"/>
  </owl:ObjectProperty>
</owl:Class>
```

Fig. 3 OWL code of the metadata format

It needs to be noted that this unified metadata format is extensible. With changes to Web documents formats, its attributes can be added.

B. Ontology Concept Format

Each ontological concept has two basic properties, which are conceptDescription and linkedMetadata.

conceptDescription is a data type property of concept, which refers to the predefined contexts that define and describe an ontological concept. It normally consists of several descriptive phases, which can be used for computing semantic similarity values with metadata (discussed in the next section).

linkedMetadata is an object property of concept, which is used to store the URIs of semantically similar metadata to the concept.

The OWL code of ontological concept format is shown below:

```
<owl:Class rdf:ID="Concept"/>
  <owl:DatatypeProperty
rdf:ID="conceptDescription">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#st
ring"/>
    <rdfs:domain rdf:resource="#Concept"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="linkedMetadata">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="linkedConcepts"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Metadata"/>
  </owl:ObjectProperty>
</owl:Class>
```

Fig. 4 OWL code of the ontological concept format

Similarly, the attributes of concepts can be added according to different ontology domains.

V. Extended Case-Based Reasoning Algorithm

As described earlier, one task of the Metadata Matcher is to compute the similarities between each ontology concept and metadata, by comparing the attribute of concept descriptions from concepts and the attribute of metadata descriptions from metadata. The computation is based on an ECBR algorithm [20]. The following is the definition of the ECBR model.

The similarity between a concept C and a metadata M is obtained by Equation (1) and (2) as follows:

$$Sim(C,M) = \underset{cd_j \in C}{Max}\left(\sum_{t_{jk} \in cd_j} \frac{f(t_{jk}, md)}{l_{cd_j}} \right) \quad (1)$$

$$f(t_{jk}, md) = \begin{cases} 1 \ if \ \exists md_t \mid (t_{jk} = md_t) \wedge (md_t \in md) \\ 0 \ otherwise \end{cases} \quad (2)$$

where cd_j is a concept description of the concept C and md is the metadata description of the metadata M, t_{jk} is a term in the concept description cd_j, l_{cdj} is the total number of terms that appear in the concept description cd_j, and md_t is a term that occurs in the metadata description md.

The principle of the ECBR model is to match between the group of concept descriptions of a concept and the metadata description of a metadata, in order to find the maximum similarity between them. The simulated scenario of ECBR matching process is represented by Fig. 5.

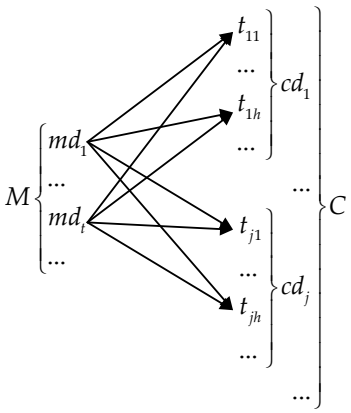

Fig. 5. Simulated scenario of the matching process of ECBR

The ECBR model is simple to implement, and it does not need to generate index terms before matching, which saves preprocessing time. It can also adapt to the frequent update of the ontologies, which often need the regenerating of index terms in most of index term-based algorithms. Since the model is independent of index terms, it does not have the issue of index term dependency.

VI. System Implementation and Evaluation

In this section, we implement the prototype of the Webpage Classification Agent and evaluate the performance of this conceptual model based on the prototype.

A. System Implementation

The whole system implementation can be divided into two tasks: 1) building the Knowledge Base; 2) implementing the Webpage Classification Agent.

For the first task, we utilize Protégé-OWL as the main tool for ontology construction. As we know, an ontology is a shared vocabulary used to model a specific domain [21], so it is not possible to design an ontology for universal domains. As a result, we must choose a particular domain for the ontology building. Here we focus on the service domain, and choose one of its sub-domains – the transport service domain as the boundary within which the ontology is built. Fig. 6 displays the screenshot of the transport service ontology in Protégé-OWL. Due to page limitation, we cannot represent the whole ontology; further information pertaining to the transport service ontology can be referenced from [22, 23].

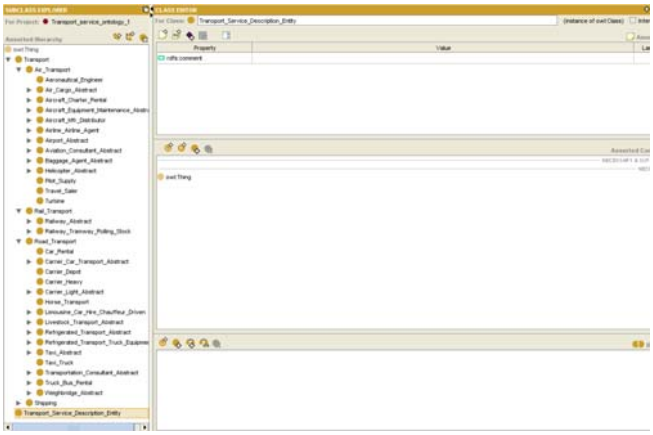

Fig. 6. Screenshot of the transport service ontology in Protégé-owl [22]

For the second task, namely Webpage Classification Agent implementation, we use JAVA as the primary tool. A multi-thread agent is implemented here. The agent can download all webpages from a specified website, and parse the contents of the downloaded webpages into pieces by the predefined parsing rules. Meanwhile, all HTML or XML tags are filtered by the agent. Following this, the agent can add OWL tags to the information pieces in order to generate metadata. Once a metadata has been generated, it will be stored into the Metadata Base, and the metadata description attribute of the metadata and the concept description attribute of the transport service ontology concept will be matched by the ECBR algorithm in order to obtain the similarities between metadata and concepts. An optimal threshold value for the ECBR algorithm needs to be chosen (which will be discussed later). If the computed similarity is beyond the optimal threshold value, the metadata and concept can be considered as similar

and linked by storing their URI to each other. Eventually, the webpages are classified by the Webpage Classification Agent.

B. Performance indicators

To evaluate our Webpage Classification Agent, Precision and Recall, two widely used performance indicators from the information retrieval file, are adopted in the following experiment.

Precision in the information retrieval is used to measure the preciseness of a retrieval system [10]. In our experiment, precision for a single concept is the proportion of linked and semantically similar webpages in all webpages linked to the concept, which can be represented with Equation (3) below:

$$\text{Precision@Single Concept} = \frac{\text{number of linked and semantically simialr webpages}}{\text{number of linked webpages}} \quad (3)$$

With regard to the whole collection of concepts, the total precision is the sum of precision for each concept normalized by the number of concepts in the collection, which can be represented with Equation (4) below:

$$\text{Precision@Whole Concept Collection} = \frac{\sum_{i=1}^{n} \text{Precision@Single Concept } i}{n} \quad (4)$$

'Recall' in the information retrieval refers to the measure of effectiveness of a query system [10]. In this experiment, recall for a single concept is the proportion of linked and semantically similar webpages in all semantically similar webpages, which can be represented with Equation (5) below:

$$\text{Recall@Single Concept} = \frac{\text{number of linked and semantically similar webpages}}{\text{number of semantically similar webpages}} \quad (5)$$

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 using Equation (6) below:

$$\text{Recall@Whole Concept Collection} = \frac{\sum_{i=1}^{n} \text{Recall@Single Concept } i}{n} \quad (6)$$

The following experiment will be made based on precision and recall.

C. Evaluation

As mentioned previously, when the ECBR computes the similarity between a concept and a metadata, an optimal threshold value needs to be chosen to determine whether or not the concept and metadata is similar. Hence, in this experiment, apart from the task of evaluating the performance of the Webpage Classification Agent, we also need to find the optimal threshold value. In order to do so, in the following experiment, we will test the performance of the agent on precision and recall, at the threshold value from 0.5 to 1, with a 0.05 increment each time.

Subsequently, we choose the Australian Yellowpages® website as the webpage source, and download 600 webpages under the category of transport from this website. Our Metadata Generator generates 1120 metadata in total from the webpages. Fig. 7 and Fig. 8 indicate the performance of the Webpage Classification Agent on Precision and Recall respectively.

From Fig. 7, it is clearly observed that the precision increases along with the increase of threshold value. The rising trend stops, and the precision remains stable in the peak when the threshold value reaches 0.8. This phenomenon is due to the fact that a higher threshold may filter more dissimilar webpages, thereby leading to greater precision. The whole variation interval for the precision is from 15.12% to 91.42%, which is a larger range.

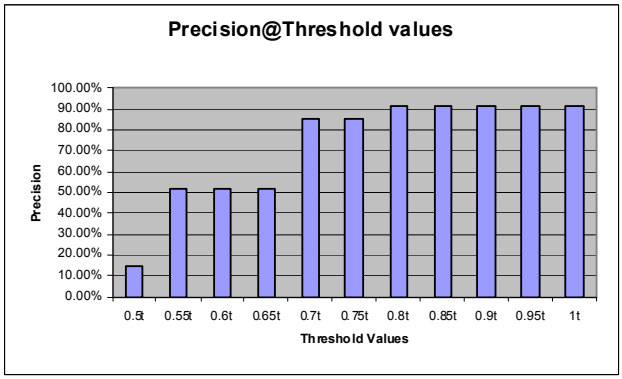

Fig. 7. Precision @ threshold values

From Fig. 8, it is found that the recall experiences several fluctuations and basically displays a falling trend along with the rise of threshold values. Eventually, the recall remains stable when the threshold value reaches 0.8. The reason for this is that higher recall values may prevent more potentially similar webpages from being linked by the ontology concepts. The whole variation interval for the recall is from 98.38% (0.55-0.65) to 99.17% (0.5), which is a tiny range compared with the precision.

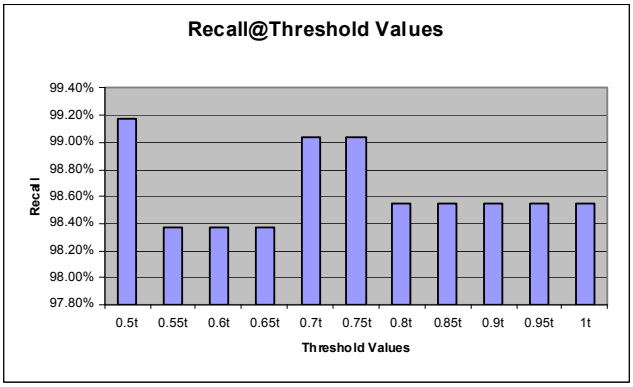

Fig. 8. Recall @ threshold values

Based on the performance of the Webpage Classification Agent on precision and recall above, the optimal threshold value is 0.8. The reason is that this value is the boundary within which the precision reaches its peak, and while the recall is lower than it is at 0.5, the gap is too tiny (0.63%) to be significant. In addition, given the performance (precision: 91.42%, recall: 98.54%) of the agent at the optimal threshold value, we can be confident that the agent delivers a superior performance in this experiment.

VII. Conclusion and Further Work

In this paper, we present an ontology-based webpage classification approach in order to assist the Knowledge Grid to classify the Web documents without semantic supports. This approach consists of a two-step process: 1) metadata are generated by extracting and annotating feature information from webpages; and 2) metadata are linked by similar ontology concepts. By making use of metadata as the intermedium between webpages and ontology concepts, the webpages are classified and semanticized by ontologies. Following this, we design the conceptual framework of a Webpage Classification Agent. The agent consists of four main parts: 1) Webpage Fetcher that can download webpages from a website; 2) Webpage Parser that parses the downloaded webpages into information pieces according to the predefined parsing rules; 3) Metadata Generator that generates metadata based on a predefined metadata format and stores them into a Metadata Base; 4) Metadata Matcher that computes the similarities between generated metadata and ontology concepts stored in a Knowledge Base, and links the metadata with similar concepts.

Subsequently we present the format of the metadata and ontology concept. Next, we present an ECBR algorithm for the similarity computation. In order to implement the prototype of the Webpage Classification Agent, we choose a sub-domain of the service domain – the transport service domain – and build a transport service ontology based on the domain knowledge, and we employ JAVA as the main tool to build the agent. With the purpose of evaluating the agent, we choose the Australian Yellowpages® website as the webpage source and undertake an experiment based on the performance indicators of precision and recall. Another task involved in the experiment is to choose an optimal threshold for the agent. From the experimental results, we find that the agent gives a persuasive performance, and the optimal threshold value is 0.8.

For ongoing and future work, we are in the process of trying other information retrieval algorithms in the Web Classification Agent, and testing it on other domains in order to obtain a better performance at a lower computing cost.

Acknowledgment

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

References

[1] J. F. Gantz, C. Chute, A. Manfrediz, S. Minton, D. Reinsel, W. Schlichting, and A. Toncheva, "The diverse and exploding Digital Universe: An updated forecast of worldwide information growth through 2011," IDC, Framingham2008.

[2] D. Boley, M. Gini, R. Gross, Eui-Hong, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "Partitioning-based clustering for Web document categorization," *Decision Support Systems,* vol. 27, pp. 329-341, 1999.

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

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

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

[6] H. Zhuge, "Communities and emerging semantics in semantic link network: Discovery and learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 21, pp. 785-799, 2009.

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

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

[9] X. Qi and B. D. Davison, "Web page classification: Features and algorithms " *ACM Computing Surveys* vol. 41, 2009.

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

[11] M. Hearst, "TileBars: Visualization of term distribution information in full text information access," in *the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, New York, 1995, pp. 59-66.

[12] A. Veerasamy and N. J. Belkin, "Evaluation of a tool for visualization of information retrieval results.," in *the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, New York, 1996, pp. 85-92.

[13] O. Zamir and O. Etzioni, "Grouper: A dynamic clustering interface to Web search results," in *the Eighth World Wide Web Conference (WWW' 99)*, Toronto, 1999, pp. 1361-1374.

[14] H. Chen, H. Fan, M. Chau, and D. Zeng, "MetaSpider: Meta-searching and categorization on the web," *Journal of the American Society for Information Science and Technology,* vol. 52, pp. 1134–1147, 2001.

[15] S. Charkrabarti, B. E. Dom, and P. Indyk, "Enhanced by hypertext categorization using hyperlinks," in *ACM knowledge discovery and data mining (KDD'98)*, New York, 1998, pp. 169–173.

[16] H. J. Oh, S. H. Myaeng, and M. H. Lee, "A practical hypertext categorization method using links and incrementally available class information," in *the 23rd annual international conference on research*

and development in information retrieval (SIGIR2000), Athens, 2003, pp. 264-271.

[17] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs: Prentice-Hall, 1988.

[18] S. Y. Lu and K. S. Fu, "A sentence-to-sentence clustering procedure for pattern analysis," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 8, pp. 381–389, 1978.

[19] O.-W. Kwon and J.-H. Lee, "Text categorization based on k-nearest neighbor approach for Web site classification," *Information Processing and Management,* vol. 39, pp. 25-44, 2003.

[20] H. Dong, F. K. Hussain, and E. Chang, "A semantic crawler based on an extended CBR algorithm," in *Lecture Notes in Computer Science: OTM 2008 Workshops*. vol. 5333, R. Meersman*, et al.*, Eds., ed Heidelberg: Springer-Verlag Berlin, 2008, pp. 1084-1093.

[21] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition,* vol. 5, pp. 199-220, 1993.

[22] H. Dong, F. K. Hussain, and E. Chang, "A transport service ontology-based focused crawler," in *The 4th international conference on semantics, knowledge and grid (SKG 2008)*, Beijing, 2008, pp. 49-56.

[23] H. Dong, F. K. Hussain, and E. Chang, "Transport service ontology and its application in the field of semantic search," in *2008 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI 2008)*, Beijing, 2008, pp. 820-824.