Automatically Attaching Web Pages to an Ontology

Robert Villa **Ruth Wilson** **Fabio Crestani**
Department of Computer and Information Science
University of Strathclyde
Robert.Villa@cis.strath.ac.uk Ruth.Wilson@cis.strath.ac.uk Fabio.Crestani@cis.strath.ac.uk

Abstract

This paper describes a proposed system for automatically attaching material from the world wide web to concepts in an ontology. The motivation for this research stems from the Diogene project, which requires the project's own databases of learning objects to be augmented with additional resources from the web. Two main approaches to this problem are being taken: one using ontology mapping, and another based on the conventional text search facilities of the web, covered in this paper. By generating queries based on the concepts in the ontology, the aim is to retrieve material from the web, and then filter it to ensure its proper correspondence with a concept. The Diogene system will be briefly outlined, before the query-generation system is described. A small pilot experiment, designed to provide some initial results and insight into the problem, is then presented.

Keywords: Ontology, information retrieval, web search, query generation.

1 Introduction

Ontologies provide a common language for sharing knowledge between members of a community of interest. In this paper we consider the problem of automatically populating an ontology with documents discovered on the web, within the context of the Diogene project[1].

Diogene is a learning broker, based on an ontology covering the Information, Communications and Technology (ICT) domain. Classified within the ontology are 'learning objects', delivered to a user in response to a request for training. The discovery of web resources to augment this collection of native learning objects is one of the key requirements of the project.

This paper will briefly outline the Diogene project and describe its 'Web Discovery' component. Two approaches to the problem are briefly outlined, before a method based on conventional web searching and information retrieval technology is presented in detail. A pilot experiment is then described, in which the presented ideas are evaluated.

2 The Diogene System

Diogene is an EC project funded under the 5th Framework Programme - Information Society Technologies (contract IST-2001-33358). Its main objective is to design, implement and evaluate an innovative web training environment for ICT professionals. The environment will be able to support learners during the whole training cycle, from the definition of objectives to the assessment of results, through the construction of custom self-adaptive courses.

At the core of the system's knowledge representation framework is an ontology covering the ICT domain, in which learning objects are classified.

Ontologies have been defined as "explicit conceptualisation[s] of a domain", in which objects, concepts and relationships between

[1] http://www.diogene.org/

them are defined as a set of representational terms, enabling knowledge to be shared and reused [4]. McGuinness discusses the spectrum of specifications which people have termed ontologies, including controlled vocabularies, glossaries, thesauri, web hierarchies such as Yahoo!, subclass hierarchies, formal instance relationships, frames, value restrictions, and logical constraints [6].

The design of Diogene's ontology [1, 7] reflects its primary use within the system: to aid the automatic creation of courses for presentation to students. It comprises a set of concepts covering the ICT domain, linked by the following relations:

- Has Part: HP ($x, y_1 \ldots y_n$) means that concept x is composed of the concepts y_1 to y_n;
- Requires: R (x, y) means that, to learn x, it is first necessary to learn y;
- Suggested Order: SO (x, y) means that it is preferable to learn x and y in this order.

Figure 1 shows a segment of the ontology.

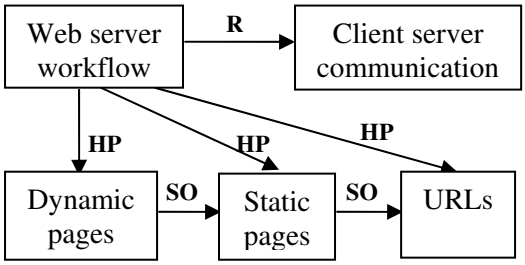

Figure 1. A part of Diogene's ontology

This means that, when a student makes a request to learn about web server workflows, the Diogene environment will specify that:

- He is *required* first to know about client server communication;
- Web server workflows comprises (*has parts*) the topics dynamic pages, static pages and URLs;
- The *suggested order* in which the student should learn the topics is URLs, static pages, then dynamic pages.

The student's prior knowledge is taken into acount at this stage, and a personal learning path is created. He is provided with material:

- That has been classified according to the concepts in the learning path;
- That matches his personal preferences (e.g. he may specify that he prefers learning from diagrams and images rather than textual documents);
- In the order specified in the learning path (successful completion of a tutor-marked test enables him to move to the next topic).

This training material may be from registered content providers, high quality and manually marked up, or free content from the web. The rest of this paper is concerned with the issues related to the discovery and classification of web material in the Diogene ontology.

3 The Web Discovery Component

To enable free web material to be used within Diogene, it must be found, classified in Diogene's ontology, and made available to the other parts of the system.

This is achieved through a Web Discovery component, in which the problem of discovering web material is being tackled in two main ways:

- Discovery by ontology mapping: by mapping external ontologies which exist on the Semantic Web to Diogene's ontology.
- Discovery on the conventional web: by searching the conventional web using automatically generated queries designed to represent individual concepts in the ontology.

These techniques are complementary. The first approach takes advantage of the new possibilities which come with formally defined ontologies. This requires an external ontology to be identified so that its attached learning objects can be attached to the Diogene ontology, and is described in a separate paper [14]. The second approach enables us to capture learning objects that are not attached to any ontology, and provides a way to access material on the current web. This approach, which will be discussed in this paper, consists in constructing a query for

each concept in the ontology, and attaching to the Diogene ontology new web pages from the results of the search.

4 Content Discovery on the Web

The content discovery system we have developed has three main steps to perform:

- Construct a query for each concept in the ontology;
- Execute that query using a web search engine; and
- Download and filter the result set from the search, to ensure a good match against the concept description [13].

This is shown in more detail in Figure 2 for a single concept and its associated learning objects. Each of these elements will be dealt with in turn in the following sections.

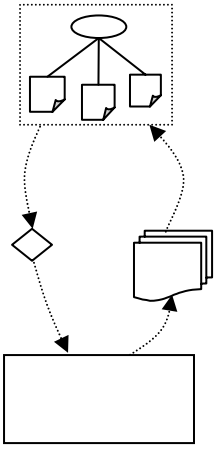

Figure 2: Web Discovery main steps, for a single concept

The motive for this work is to populate our ontology, so a primary consideration in designing this system was to be open to incorporating as many web resources as possible. As such, our techniques involve "flattening" the ontology, disregarding the Has Part, Suggested Order and Requires relations, and instead focusing on the individual concepts, essentially treating them as independent category labels.

Our approach, which brings together existing techniques, puts a very low computational load on the system and is applicable to highly heterogeneous documents, so that large quantities of new material can be acquired relatively easily.

4.1 Generating Queries

There are three main sources of textual information in Diogene which can be used to describe the concepts in the ontology:

- Concept name;
- Concept description;
- The learning objects attached to the concept.

All concepts have a name, and may also have a description or learning objects attached. Names, however, are short (they are labels comprising 4-5 word phrases). Concept descriptions are also short, typically a sentence or two.

The textual content of the learning objects provides a larger and richer source of information than the names and descriptions, and we have decided to begin by using this source of data, comparing it with the concept name as a baseline.

Queries are generated from the learning objects using text processing techniques which are in common use in Information Retrieval [10, 12]. For each concept, the attached learning objects are loaded, and the text extracted and amalgamated into a single unformatted document. Since web search engines, such as *Google*[2], have an upper limit on the length of a query (ten terms in *Google*'s case), the resulting text document must be summarised.

The set of text documents for all queries can be considered as a single collection, and be indexed as such using standard Information Retrieval techniques. This involves separating the text into tokens, removing common terms, and weighting the terms using the Term Frequency – Inverse Document Frequency (TF-IDF) measure. The

[2] http://www.google.com

result of this process is a separate term-weight vector for each concept in the ontology (this vector will be used later in the filtering stage, section 4.3). To generate the final query, the *n* terms with the largest weight in each vector are selected. Stemming [9] of words is not currently used, to ensure that the query contains only complete words.

4.2 Web Search

Once the query has been generated, a web search can be carried out using any of the current web search engines available. In our current implementation we are using *Google*, since this particular engine has a non-commercial SOAP interface making its use relatively easy in a non-interactive environment. Other search engines, however, may be used.

The result of a web search is a list of hypertext links to potentially relevant material. This list is then passed to the next stage of the process, which filters the results.

4.3 Filtering the Results

Filtering the results of the web search may at first appear extraneous: if the search query is built from content associated with a particular concept, will the results not reflect that concept? Some filtering of the results will probably be useful, however, for the following reasons:

- It is likely that some links returned will be inactive. Any broken links must be removed.
- The content of the web page may have altered since it was indexed by the search engine.
- The web search is based on only a short query. In the filtering stage all information about the concept can be used to decide on the relevance of the web page to the concept, including the full text of the learning objects.

This stage can be likened to the problem of text filtering [2, 11]. From [11]:

> "A text filtering system sifts through a stream of incoming information to find documents relevant to a set of user needs represented by profiles."

While complex categorisation and filtering techniques could be used on the results of the web search to identify relevant learning objects [13], at this stage we decided to use a simple technique in order to test the feasibility of the overall approach. In our system the incoming information is the result of a web search, and rather than 'user needs' we filter based on concept information. Document vectors are constructed from the web documents, using techniques similar to those for constructing queries. Then, the similarity between the resulting web document vector and the vector produced as a by-product of query creation (section 4.1) is computed using the cosine measure [10]:

$$sim(\mathrm{W_d}, \mathrm{W_c}) = \frac{\sum_{i=1}^{n} w_{di} w_{ci}}{\sqrt{\sum_{k=1}^{n} w_{dk}^2} \sqrt{\sum_{j=1}^{n} w_{cj}^2}}$$

Where $\mathrm{W_d}$ and $\mathrm{W_c}$ are the document vectors for the web document $<w_{d1}, w_{d2}, \ldots w_{dn}>$, and concept representative $<w_{c1}, w_{c2}, \ldots w_{cm}>$, respectively.

The similarity values generated by this measure can then be used to decide whether an external web page should be imported into Diogene or not. The most straightforward way to achieve this is to set a threshold. This is an acceptable solution for the purposes of Diogene, which is intended for learners, only if the final users are not required to do anything to set the threshold. This is a strong constraint on which we are currently working. We are aiming at using users' judgements to set the threshold in a dynamic and adaptive way.

Our priority is precision rather than recall, and a technique which is 'absolute' is preferable to one which is relative to the quantity of material retrieved. For example, selecting the *n* most

similar web pages to the concept (with *n* remaining static for whatever web material and concept), therefore attempting to reproduce a 'best available' strategy, may not be acceptable for the users of the system. More complex text filtering solutions, such as those presented in [2], may in the future replace the simple scheme outlined above.

4.4 Integrating Web Material into Diogene

The final stage of the web discovery process is the extraction of markup from the web resource, and storage of the web link in a local store. While we have used the term 'attach' to describe the process of linking a Diogene concept to a web resource, in practice there is no explicit link between concept and resource. Instead, the resource is recorded in the Web Discovery's database, and provided to the other web components in Diogene when requested. For example, when an external component requests all material about a concept '*x*', the web material known to the Web Discovery component will be returned. This allows the local store to be dynamically altered to reflect changes in the web, such as a web page disappearing, and changes in the filtering threshold, removing material deemed not fully relevant by a dynamically changing threshold. The automatic markup of the web page must also be executed at this stage, but will not be covered here since it does not impact on the knowledge discovery process itself.

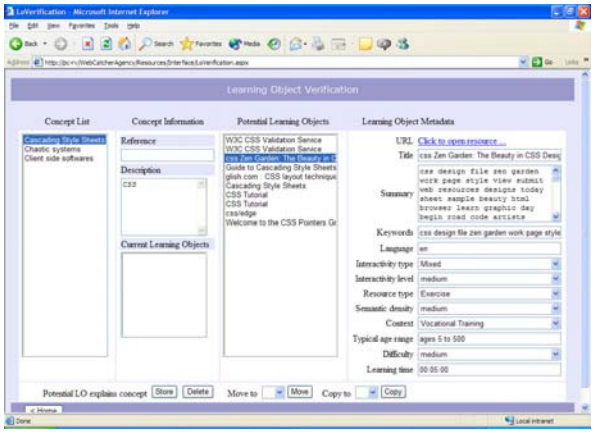

Figure 3. Content discovery interface

5 Pilot Experiment

To provide an initial indication of how such a web discovery system might perform, a small pilot experiment was set up. At the time of writing, Diogene's ontology was still in production; in its place, we used the ACM Computing Classification System (CCS). Our ontology is based on the CCS, and the two systems have highly similar coverage and granularity. The notable difference is that the CCS is a hierarchical scheme, whereas the ontology employs more complex relations; however, since our techniques ignore any links between terms, the CCS can be processed in much the same way as the ontology, providing useful feedback on our methods (the labels in the CCS were used as "concept names").

For the purpose of generating concept descriptions and evaluating our algorithms, a collection of documents classified in the CCS was used: *Journal of the American Society for Information Science* (1996-2000), *Artificial Intelligence* (1984-2002), *Computer Networks* (1999-2003), *International Journal of Human-Computer Studies* (1994-2003), and *Information Processing and Management* (1984-2003).

This collection was then split into two equally sized parts. One half was used to represent Diogene's ontology ("ontology collection"), and the other half to represent the web ("web collection"), from which we want to acquire new documents.

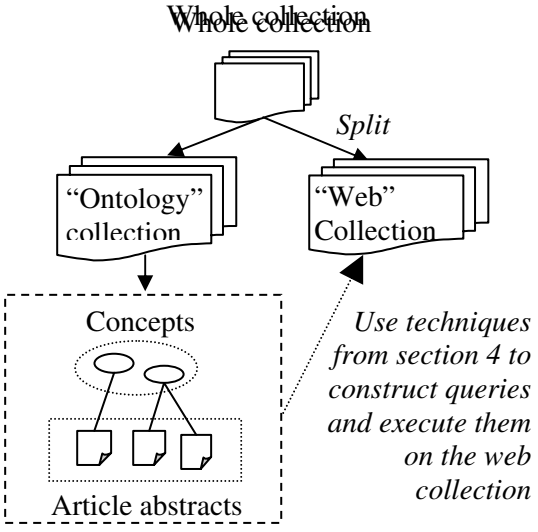

Figure 4: The experimental setup

With this experimental setup, it was relatively easy to generate the 'correct' set of documents from the web collection, which we hoped the information retrieval engine would find for a concept in the ontology collection. There are a number of things to note:

- Only the text of the document abstracts was used to create the queries.
- The abstract text was used to represent the documents in the web collection.
- Instead of a web search engine, the Lemur information retrieval engine was used [8]. TF-IDF weighting was used.

A subset of the CCS was used, containing only those category labels ("concepts") which had associated documents from the ontology collection. The assumption was that users of the Diogene system would only request new documents for concepts already in use. To provide a balanced test, we removed CCS labels from the web collection which did not exist in the ontology collection, and vice-versa. In this way, we could ensure all concepts in the ontology would have at least one relevant document in the web collection. Some statistics are provided in tables 1 and 2.

Table 1: Ontology Collection statistics

Number of concepts	123
Average length of concept name	3.8 words
Average length of text in all documents per concept	2094.1 words
Average number of learning objects per concept	13.2

Table 2: Web Collection statistics

Number of documents	1189
Average abstract length	147.8 words
Total size of collection	1.28 MBytes

No filtering stage was used in this pilot: in carrying out the experiment, it was hoped to gain an idea of the capabilities required by the filtering stage of the system.

For the purposes of the experiment, the length of all queries generated from the concepts was set to 10 terms, to match the maximum allowed by *Google*. Four retrieval runs were executed:

1. Queries automatically generated from all concepts. In this case a query was generated even if the concept in question had only a single learning object attached.
2. Queries automatically generated, but only from concepts with 3 or more documents attached.
3. Queries automatically generated, but only from concepts with 6 or more documents attached.
4. Concept name used as query.

For runs 1 and 4, all concepts in the ontology collection were used for retrieval. For run 2, only 70 of the concepts had more than 3 documents attached, therefore only 70 queries were generated for testing. Similarly for run 3, only 44 queries were generated. Runs 2 and 3 were executed to gauge the number of documents which might be required in Diogene for a good query to be generated: the hypothesis was that more documents would provide a better description of the concept, which we hope would lead to a better query.

Figure 5 shows precision for each of these runs, at the eleven standard recall points used in information retrieval evaluation [3].

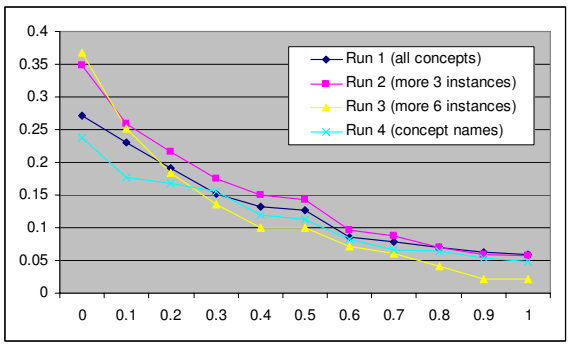

Figure 5: Interpolated precision values at 11 recall points, for the four evaluation runs

These results have been placed on a single graph for convenience, although only runs 1 and 4 are

directly comparable - runs 2 and 3 used different numbers of queries.

As can be seen, the precision results are low. There is a slight improvement when only considering concepts with more than 3 or 6 documents attached, as would be expected. Using concept names as queries also produces low results.

As a comparison, we also ran these same tests against the *full* collection of documents (ontology and web collections together). The results of these four runs are shown in Figure 6. As would be expected, the results for run 4 (using concept names for queries) are relatively uniform. The results for the other runs increase dramatically in precision.

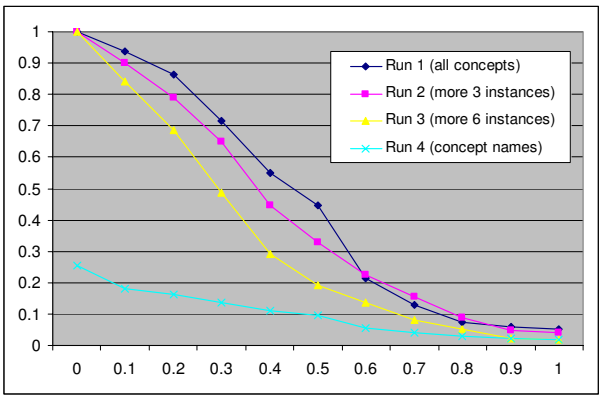

Figure 6: Interpolated precision values at 11 recall points, retrieval from the whole collection.

6 Conclusions

From the results of this pilot, it may be concluded that the approach of extracting automatically generated queries from documents attached to concepts is not likely to produce hugely relevant results lists from the queries.

While low precision results may be expected, what is more surprising is the similarity between the performance of the different runs, and the behaviour of the interpolated precision graphs. This may be due to the makeup of the test collection – many of the documents indexed came from similar areas (e.g. *JASIS* and *IP&M* have similar coverage). In addition, each of these documents have been manually classified by authors in relatively few places within the ACM CCS. The documents may be relevant, or partially relevant, to many more concepts. Our automatically generated relevance judgments are based on where authors have placed their own work, and not on explicit relevance judgments by multiple human classifiers.

Although the automatically generated queries show a slight improvement over using concept names only, it could still be concluded that the concept name is as good a query to use in a web search as an automatically generated query, given the restriction on the length of queries which may be run by search engines such as *Google*, and the extra effort required to produce the queries.

It must be noted that, while the ACM CCS provided a good representation of the concepts in Diogene's ontology, the documents used in our experiment were abstracts of scientific articles rather than the learning objects in our system. Future research will investigate the effects of the type of material used to generate queries, however we feel that article abstracts were a reasonable substitute since they share some properties of learning objects – namely, they are short, concise, focused and self-contained.

What these results do emphasize strongly is the importance the filtering stage will play in any similar technique. Indeed, multiple queries, despite the lack of precision, may be used to discover different web resources. Having an efficient and reliable method of checking the results of searches against concepts would appear to be vital if such an approach is to work.

7 Current and Future Work

Further work is currently being undertaken on explaining the current results, both by investigating the nature of our test collection, and the automatically generated relevance judgments.

In addition, the focus of some of our efforts has moved from constructing 'good' queries to filtering the query results. The techniques we are using are based on the most current and accepted results of text categorisation [13] and

filtering [5]. However, it should be noted that for the final version of the Diogene system we envisage a semi-automatic approach to the filtering and attachment of web material to the Diogene ontology. In fact, it is necessary that a human checks the correct filtering and classification of this material, since 100% precision is essential for our users' satisfaction. While a good filtering system will considerably aid this process, we do not believe it will ever be fully automatic.

One place where automatically generated queries will still be required, however, is in web searches in multiple languages. In Diogene's current ontology, all names and descriptions are in English only. If we are to aim to find resources on the web in other languages, queries in the target language may still be generated using the technique described in section 4.

The Web Discovery component is currently under development, with plans for further evaluation of the component individually, and within the context of the wider Diogene system.

Acknowledgements

This research was supported by the Information Society Technologies project Diogene (IST-2001-33358).

References

[1] Capuano, N., Gaeta, M., Micarelli, A. and Sangineto, E. (2002). An integrated architecture for automatic course generation. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, Kazan, Russia.

[2] Foltz, P. W. and Dumais, S. T. (1992). Personalized information delivery: an analysis of information filtering methods, In *Communications of the ACM*, 35 (12).

[3] Frakes, W.B. and Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*, Prentice Hall.

[4] Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5 (2).

[5] Losee, R. M. (1998). *Text Retrieval and Filtering: Analytical Models of Performance*. Kluwer Academic Publishers.

[6] McGuinness, D. (2002). Ontologies come of age. In *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential.* Fensel D., Hendler J., Lieberman H. and Wahlster, W. (eds), MIT Press.

[7] Nikolov, R., Stefanov, K., Vladinova, L. (2003). Professional e-learning: technological standards, methodological challenges and advanced applications. In *Proceedings of the International Conference on New Technologies in Learning*, Bulgaria, Sofia.

[8] Ogilvie, P. and Callan, J. (1983). Experiments using the Lemur toolkit. In *Proceedings of the Tenth Text Retrieval Conference (TREC-10).*

[9] Porter, M.F. (1980). An algorithm for suffix stripping. *Program* 4 (3).

[10] van Rijsbergen. K. (1979). *Information Retrieval.* London: Butterworths

[11] Robertson, S, and Soboroff, I. (2002). The TREC 2002 Filtering Track Report. In *The Eleventh Text Retrieval Conference..*

[12] Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company

[13] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34 (1).

[14] Villa, R., Wilson, R. and Crestani. F. (2004). Ontology mapping by concept similarity. Accepted for *International Conference on Digital Libraries (ICDL 2004)*. New Delhi, India.