

Adaptive Automatic Classification on the Web

Charlotte Jenkins

*School of Computing, Information Systems & Mathematics, South Bank University
103 Borough Road, LONDON, SE1 0AA, UK
jenkincm@sbu.ac.uk*

Dave Inman

dave@sbu.ac.uk

Abstract

This paper explores the motives and issues associated with automatic classification on the Web. Experimental software tools are introduced which enable the automatic generation of vocabularies, representing any given classification scheme, using pre-classified documents. An internal representation of the required classification hierarchy is automatically generated and subsequent documents are filtered through this hierarchy and automatically classified. The flexible, dynamic manner in which the classification vocabularies are defined creates an environment in which different collections of documents can be classified according to different, extensible classification schemes. Evolving, changing vocabularies and hierarchies can be dynamically generated and modified over time.

1. Introduction

The earliest tools for resource discovery on the Web evolved from 'hot lists' into classified directories in around 1993. These tools provided manually defined lists of links organised hierarchically under subject headings. Classified directories, many of which are still popular (e.g. Yahoo![1] receives 50 million queries per day [2]), have the benefit of manual classification and resource description. They tend to be very accurate and provide access to high quality information, often with accompanying reviews. The use of a classification hierarchy gives the user the option to query or to browse for relevant information, refining their search incrementally in an interactive manner if preferred. The limitations of classified directories however, are a consequence of their manual maintenance; they are totally incapable of keeping up with the pace of change on the Web and are often incomplete and generally out of date.

The idea of automating the resource discovery process emerged in 1994 when tools such as the World Wide Web Worm [3] were developed. The original idea behind automated search engines was to improve Web coverage by deploying robots to follow hyperlinks from one page to the next in order to retrieve and index the entire Web.

Automatic full text indexing enabled user queries to be matched against the full text of the documents instead of just manually selected keywords. This automated approach, however, suffered from a lack of limitation. Users were inundated with thousands of results not necessarily sharing the same subject matter and often the few relevant hits were lost among many irrelevant ones. Search engines have gradually become more sophisticated and those that have survived and maintained popularity (Alta Vista [4], Infoseek [5], Lycos [6], Excite [7], Hot Bot [8] etc) use Information Retrieval techniques such as the Vector Space Model [9] for clustering documents that share a high coincidence of significant terms. The significance of a term is often calculated using its Inverse Document Frequency (IDF). Although the performance of the major search engines has improved in recent years, most users would agree there is room for further improvement and recent research suggests that this may come in the form of automatic classification.

Automatic classification could combine the advantages of directories and search engines by enabling automated tools to organise resources by subject matter without compromising their ability for automatic resource discovery and indexing. Automatically organising resources according to a classification ontology has many benefits:

- An improved notion of context can be obtained when documents are organised according to subject matter. It is more likely that the results of a query will be focused on documents sharing relevant subject matter.
- Users can be given the option to enter a query or browse the hierarchy incrementally. Although most automated search engines also provide a browsable directory, this is a manually classified subset of the resources found in the automatically generated index. Automatic classification would enable the generation of comprehensive classified directories for the user to browse through if preferred.
- Queries can be matched against *class vocabularies* representing each node of the classification hierarchy, enabling the search engine to suggest relevant subject areas before actually generating results. Improved

interaction with the user could enable the acquisition of better focussed queries which in turn could result in more accurate results.

- The class vocabularies representing each node of the classification hierarchy could be automatically generated using pre-classified documents in different languages, enabling the classification and retrieval of multilingual documents. Queries in one language could retrieve multilingual documents all sharing the same subject matter.
- Automatic classification can be used to extract context sensitive metadata describing classified documents[10].

2. Issues surrounding Automatic Classification

Automatic classification is not a new concept where traditional Information Retrieval (IR) is concerned. Fairthorne published *The mathematics of classification* in 1961 [11]. Recent research has explored the potential and possibility of automatic classification on the Web:

- The Scorpion project [12] of the On-line Computer Library Center (OCLC) uses the Electronic Support System (ESS) records, which are used to generate the definitive Dewey Decimal Classification (DDC [13]) guide, as a knowledge base for automatic subject assignment.
- The European Union DESIRE project [14] has published results on an automatic classification system that classifies engineering documents using a specific engineering thesaurus to provide the structure and vocabulary for the classification ontology.
- The classifier associated with the Wolverhampton Web Library (WWLib) [15] automatically classifies Web documents according to DDC using manually defined class representatives.

All three of these classifiers use manually defined vocabularies to classify documents according to a pre-defined classification scheme. Traditional IR classifiers tend to use pre-classified training documents to automatically acquire the necessary class vocabularies.

2.1. Manual Vs Automatic Vocabulary Definition

A class vocabulary (often called a class representative) is a set of terms which represent that class or that node of the hierarchy. The terms making up the vocabulary are those thought to be important (by a human) or those that are found to have a high coincidence (by machine) in documents belonging to that class. There are very good reasons for manually defining the vocabulary for a Web classifier.

- Web documents are very different to those documents found in the corpus of traditional IR systems. They are often very short with little (or no) descriptive text and can have a tendency to contain 'misleading' vocabulary terms. Automatically classifying Web documents using good, accurate class vocabularies is a difficult task, using such documents to actually generate the class vocabularies in the first place is potentially hazardous.
- In at least two of the above automatic classification projects, the documents were classified according to a very well defined traditional classification scheme, the classes of which can be very precise (finely grained) and have very specific vocabulary terms. Relying on Web documents to highlight these terms is perhaps optimistic.

There are, however, obvious disadvantages associated with the manual definition of vocabularies:

- It is a tedious, laborious, slow process requiring the input of universal subject experts to be done properly.
- It results in a static, inflexible vocabulary that cannot easily be updated.
- It usually results in a uni-lingual vocabulary with no obvious, reliable method of translation.

By the same token, there are good reasons for pursuing automatic vocabulary definition:

- Dynamic, evolving, flexible vocabularies can be created and maintained using the vocabularies of pre-classified documents.
- The vocabulary learning process can be on-going with class vocabularies being modified each time a new document is found to belong to that class.
- The structured nature of HTML can be used to help identify important terms such as those that occur in titles or headings.
- Multilingual training documents can be used to create multilingual class vocabularies, enabling the automatic classification and retrieval of multilingual documents.
- If vocabularies are automatically generated the classifier need not be tied to one particular classification scheme. This means that the same set of software tools could be used to classify different collections of documents according to different classification schemes.

2.2. Classification Schemes

The disadvantage of having a classifier that is tied to one classification scheme is that the scheme might not be appropriate for every application or for every collection of documents. The Resource Organisation And Discovery in Subject-based services (ROADS [16]) project identified problems when trying to merge their subject gateways

because different subject areas were better suited to different classification schemes.

A survey of different classification schemes on the Web [17] carried out in May 1997 by the UKOLN Metadata Group as part of the European DESIRE project, found seventeen Web sites claiming to use DDC to classify resources, five claiming to use the Universal Decimal Classification scheme (UDC), five using the Library of Congress Classification scheme (LCC) and various others using "National General Schemes" or "International subject specific schemes". This, of course, does not cover all those sites which use their own classification scheme which is probably the most common option on the Web.

Although traditional classification schemes, such as DDC are proven in book libraries, there is no evidence to suggest that they are the best option on the Web. In fact, it could be argued that the distribution of information on the Web, in terms of subject matter, is totally dissimilar to the distribution of information in print and therefore a purpose built contemporary scheme (such as that devised by Yahoo!) is more appropriate. Before Melvil Dewey invented the DDC scheme in 1876 [18] each library organised books according to available shelving and when a category out grew its shelf, the whole scheme had to be altered. The idea of assigning each book to an intellectual space in a classification hierarchy rather than to a physical space on a shelf was a significant revelation for book libraries. Shelving is not such a crucial concern for Web libraries but it could still be argued that assigning resources to a standard scheme would encourage interoperability and information sharing across tools for resource discovery. This could conceivably also assist users in knowing where to look for required information within the scheme. Perhaps what is required is a new standard for the Web, a more extensible scheme that has loosely defined general categories which could be refined lower down the hierarchy in a manner best suited to the application.

3. An Experimental Vocabulary Generator

Software tools that enable the automatic generation of class vocabularies representing any given classification scheme are potentially useful for a range of applications. The experimental software tools described here have been developed primarily to assist the development of a multilingual search engine that could be used as an aid in learning a new language. The intention is that it will be possible for queries in one language to retrieve multilingual documents sharing the same subject matter. The flexible nature of these tools, however, means that they are potentially useful for a range of applications.

Class vocabularies (representing each node of the classification hierarchy) are automatically generated by

merging the vocabularies of given, pre-classified training documents and then extracting the terms thought to be particularly important (those that appear frequently, those that occur over the widest range of training documents, those that occur in titles and headings) in order to generate the 'final' vocabulary for the class. A series of URLs pointing at the pre-classified training documents for a particular class can be read from a file (referred to below as the training file), the name of this file and the name of the corresponding class are passed to the system by the user.

3.1. Document Analysis



Figure 1. A document vocabulary vector, each element of which contains a DocEntry

Each URL in the training file is parsed to generate a document vocabulary vector as shown in figure 1. The vector comprises a series of DocEntry objects (shown in figure 2) each of which comprises a word, its frequency (within the document) and flags indicating whether it was found in the title or a heading. The vocabulary vector is generated as follows:

- Use the URL to open a connection to the document
- Extract each line from the document
- For each line, remove the HTML inserting markers where <TITLE>, <H1> or <H2> tags were found.
- Analyse each remaining word in each line and:
 - If it is a <TITLE>, <H1> or <H2> marker (beginning or end) set local flags (to true or false) to indicate whether the words to follow are title or heading words.

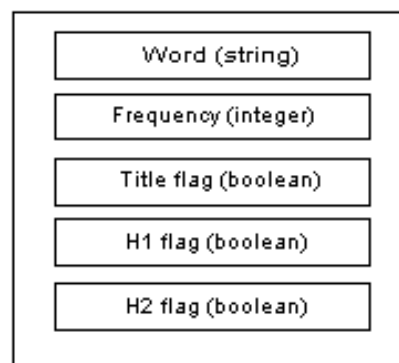


Figure 2. A DocEntry object used to store each unique word found in the document.

- If it is a stop word (currently stop words are read from a manually defined list) do nothing
- If it is a word that already exists in the document vocabulary (shown in figure 1) increment the frequency and set the title, H1 and H2 flags (according to the state of the local flags) on the appropriate DocEntry (shown in figure 2).
- Otherwise generate a new DocEntry object with a frequency of 1 and set the title and heading flags accordingly. Add this DocEntry to the document vocabulary.

When the entire document has been parsed to produce the document vocabulary vector, this vector can then be used to edit the class vocabulary.

3.2. Editing the Class Vocabulary

Each time a new training document is parsed, the resulting vocabulary vector is used to edit (or create) the class vocabulary vector as shown in figure 3. This is done using the following procedure:

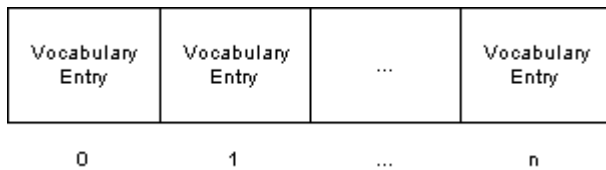


Figure 3. Class vocabulary vector, each element of which comprises a VocabularyEntry object

- Take each word (DocEntry) from the document vocabulary vector.
- If the word already exists in the class vocabulary (shown in figure 3), edit the appropriate VocabularyEntry (shown in figure 4) by incrementing the document frequency, increasing the overall frequency (by the amount specified in the frequency of the DocEntry) and incrementing the title, H1 and H2 frequencies if the corresponding flags are set in the DocEntry.
- Otherwise generate a new VocabularyEntry (shown in figure 4) for the word, set the overall frequency to equal the frequency in the DocEntry, set the document frequency to 1 and set the title, H1 and H2 frequencies to 1 if any of the corresponding flags are set in the DocEntry.

Applying this process to each training document results in access to the following information:

- All the words that were found in the documents.
- The overall frequency with which each word occurred throughout all the documents.
- The number of documents in which each word occurred.
- The number of documents in which the word occurred in the title.

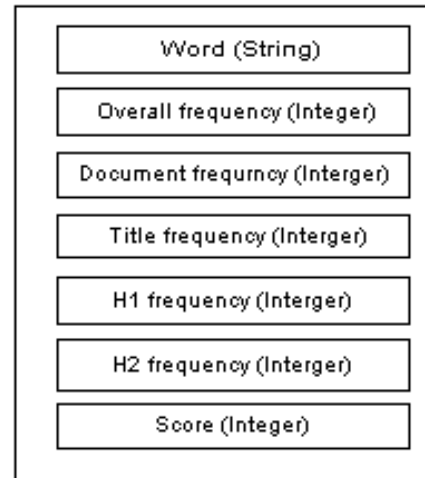


Figure 4. VocabularyEntry object used to store each unique word in the class vocabulary and its associated frequencies

- The number of documents in which the word occurred in an <H1> heading.
- The number of documents in which the word occurred in an <H2> heading.

3.3. Creating the Final Vocabulary for a Class

When all the available training documents have been parsed, a 'final' vocabulary is then generated. This is done by extracting all those words that occurred in more than 10% of the training documents and then scoring each word and extracting the highest scoring 50%. Scores are calculated as follows:

$$score = f(d + t + H + h)$$

where:

f = the overall frequency

d = document frequency (the number of documents in which the word occurred)

t = title frequency (the number of documents in which the word occurred in the title)

H = <H1> heading frequency

h = <H2> heading frequency.

The resulting final vocabulary vector is saved to a simple text file where it can be edited manually if required. The initial complete vocabulary is also saved so that further training and refinement can be carried out with more training documents if required.

4. Generating the Hierarchy

In order to automatically classify documents, the automatically generated vocabularies need to be organised into a classification hierarchy. To generate this hierarchy, the system needs to know which vocabulary belongs in which node of the hierarchy. Currently this is done using

a numerical prefix on the class name that is presented to the system along with the name of the training file. Two names are actually given:

1. **Node Name** - The node name has the form *<numerical prefix>-<classification scheme name>*. The numerical prefix indicates which node of the hierarchy this vocabulary represents using the convention shown in figure 5. This only allows there to be nine sub-nodes of each node but if necessary this could be extended using alphanumeric characters. The classification scheme name identifies which scheme the node belongs to (it is possible to define multiple classification schemes) and is the same for every class vocabulary of a particular hierarchy. The final vocabulary for each class (node) is stored in a simple text file called: *final-<numerical prefix>-<classification scheme name>.txt*
2. **Given class name** - The given class name is a textual label used for human differentiation when documents are automatically classified. Example class names might be "Renaissance Literature" or "Plant Biology". The class name is written to/read from the first line of the vocabulary text file.



Figure 5. Classification hierarchy showing the numerical prefix naming convention

When the classifier is run the hierarchy is built by firstly generating an instance of the root node, like that shown in figure 6. The vocabulary for the root node is generated using the *final-0-<classification scheme name>.txt* file (the classification scheme name is specified as a command line argument when the classifier is run). Its sub-nodes and their sub-nodes (etc., etc.) are generated recursively by appending digits to the numerical prefix and forming the appropriate vocabulary vectors. Node objects are then assigned to a vector of sub-nodes within the parent node (see figure 6). If no vocabulary file can be found for a particular node the node is not formed. VocabularyEntries can be merged upwards in the hierarchy so that the sub-nodes of the root node contain a subset of the root vocabulary and their sub-nodes contain a subset of their vocabulary and so on.

The automatically generated hierarchy can then be used to automatically classify further documents.

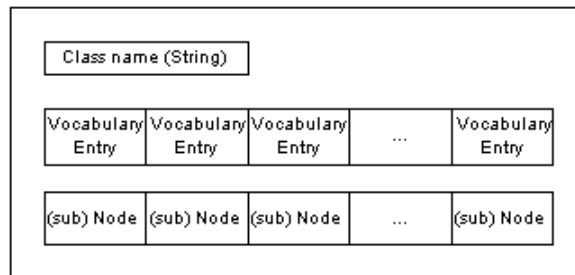


Figure 6. A node object comprising the class name, the vocabulary vector for that class and a vector of sub-nodes (each one an instance of a node object).

5. Automatic Classification

In common with the WWLib classifier, mentioned in section 2, this classifier works by recursively comparing documents with the nodes of the classification hierarchy. Only the sub-nodes of those nodes where the class vocabulary was found to have a significant measure of similarity with the document vocabulary are pursued. This means that only potentially relevant branches are pursued and the classifier does not waste time comparing the document with classes where it is unlikely that there will be significant similarity.

When documents to be classified are presented to the system they are parsed to produce a document vocabulary vector in exactly the same way as the training documents are processed (as described in section 3.1). Each DocEntry in the document vocabulary vector is then compared with VocabularyEntries in the nodes of the classification hierarchy, beginning with the root node. Each time a word in the class (node) vocabulary matches a word in the document vocabulary the score associated with the class VocabularyEntry (assigned to the class VocabularyEntry during the generation of the final vocabulary - see section 3.3) is added to a score associated with the document vocabulary DocEntry. This DocEntry score is calculated as follows, where *f* = the frequency from the DocEntry (i.e. the frequency with which the word occurs in the document):

$$score = f$$

if the word occurred in the title:

$$score = score + (f * 2)$$

if the word occurred in an H1 heading:

$$score = score + f$$

if the word occurred in an H2 heading:

$$score = score + \left(\frac{f}{2}\right)$$

The result of adding the DocEntry and the class VocabularyEntry scores together is then added to a total score for the doc/class comparison. When each of the

words in the document vocabulary have been compared with each of the words in the class vocabulary, the total score associated with that comparison is then checked for significance using the Dice Coefficient [19]:

$$2 \frac{X \cap Y}{X \cup Y}$$

where:

$$X \cap Y$$

represents the total score associated with the comparison and:

$$X \cup Y$$

represents the length of the document vocabulary plus the length of the class vocabulary.

Using this coefficient allows the relative lengths of the two vocabularies to be normalised. A 'significant point' can be set between 0 and 1 and any comparison scoring higher than the specified significant point is then considered significant and the document may continue to be compared with any sub-nodes. When a significant match is found the classification (textual label) and the comparison score is stored with the document vocabulary (as part of a document object) and then the highest scoring classifications are presented as being the appropriate classifications for the document.

6. Evaluation

The classification system was tested using a collection of architecture related documents. Architecture was chosen because it is thought to be an interesting subject area from a multilingual/ multinational perspective (as mentioned above, this classifier is intended to assist the development of a multilingual search engine that could be used to assist in learning a new language).

For evaluation purposes, vocabularies were based on a subset of those found under the Yahoo! category Arts/Design_Arts/Architecture. Figure 7 shows the

classification hierarchy that was automatically generated.

The vocabularies were generated by taking pages of links from Yahoo!, stripping out everything except the URLs and then passing those URLs to the system in a training file. The system then automatically acquired the contents of each document using the URLs, and generated document vectors for each one. Each document vector was then used to build a vocabulary vector for the specified node vocabulary as described in section 3.

The results of automatically classifying a selection of documents belonging to the categories shown in figure 7. are shown in Appendix A. Two documents were randomly selected from each Yahoo! category corresponding to a leaf node (node with no sub-nodes) in figure 7 in order to generate the test corpus (28 documents in total). Each classification was scored according to how many correct paths (from node to sub-node) were followed out of the total number of paths needed to reach the correct node. If a wrong path was followed (to the wrong sub-node) a point was deducted. The classifier scored 61 points out of a possible 80 resulting in 76.2% accuracy. Four of the test URLs were 'Not Classified', in each case this was due to the document containing very little descriptive text and consequently not acquiring any significant word matches. Unsurprisingly, documents were more accurately classified if they belonged to nodes where the vocabulary had been generated using a larger selection of pre-classified documents. Where possible, approximately 20 training documents were used to generate each vocabulary, but in some cases that many documents were not available.

7. Conclusions

Automatic classification has the potential to empower automated tools for resource discovery with the ability to organise and retrieve collections of documents sharing the same subject matter. This could be used to assist users in

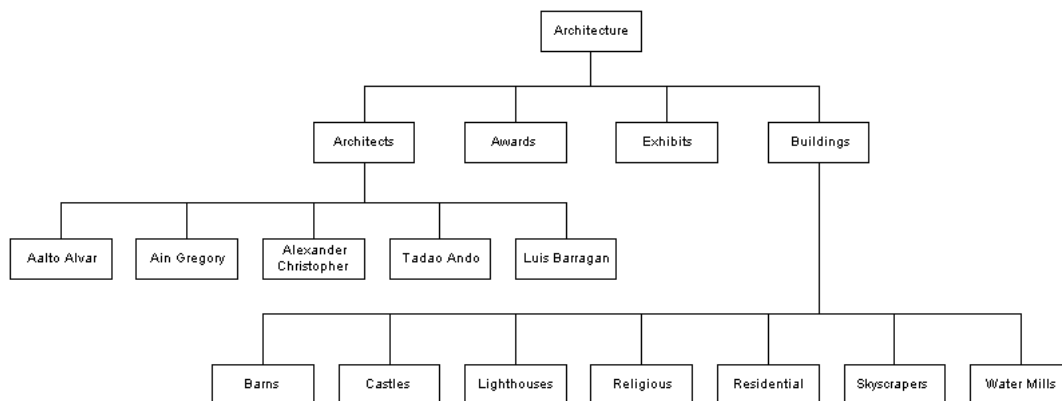


Figure 7. Classification Hierarchy for Architecture related documents based on Arts/Design_Arts/Architecture from Yahoo!

refining their queries in addition to adequately focussing results. Comprehensive classified directories could be generated providing the user with the option to browse through the entire corpus of an automated search engine if required. Improved access to multilingual information is also a possibility.

The software tools described in this paper enable the automatic generation of the necessary vocabularies for an automatic classifier and the automatic generation of the corresponding hierarchy. This can be used to classify collections of documents according to any classification scheme and the vocabularies can be amended and improved over time without the need for laborious manual input or without necessarily having to consult subject experts or to wait for thesauri to be updated.

Future developments include:

- Statistically generating stop lists - currently stop lists are read from a manually generated file but it would be advantageous, particularly from a multilingual perspective, to automatically stop words with dramatically high frequencies.
- Implementation of Natural Language Processing techniques - it is hoped that the system will be enhanced so that it will add extra importance to non-ambiguous nouns and/or to words frequently identified as holding the subject of a sentence etc. Improved stemming facilities are also required.
- Implementation testing over a wider, deeper hierarchy.
- Implementation testing using multilingual documents.
- Automatic metadata generation describing documents, class vocabularies and whole classification schemes.
- Implementation of an automatically generated hierarchy to develop a search and browse interface to facilitate retrieval.

References

1. Yahoo! <http://www.yahoo.com>
2. J. Pedersen and W. Chang, *Inside Internet Search Engines*, tutorial session, SIGIR, Berkeley, California, August 1999
3. O. McBryan, *GENVL and WWW: Tools for Taming the Web*.

Appendix A

The table below shows the results of automatically classifying a test corpus (two randomly selected URLs from each Yahoo! category corresponding to a leaf node shown in figure 7) using the automatically generated classification hierarchy.

Title	URL	Yahoo! category	Automatic classification result	Score
MoMA exhibitions Alvar Aalto	http://www.moma.org/exhibitions/aalto/index.html	Architecture/Architects/Masters/ Aalto__Alvar/	Architecture/Architects/Aalto Alvar	3 / 3
Alvar Aalto's Säynätsalo Municipal Centre	http://www.acs.ucalgary.ca/~rmc/cuaig/art_425/aalto/index.html	Architecture/Architects/Masters/ Aalto__Alvar/	Architecture/Architects/Aalto Alvar	3 / 3

4. Alta Vista <http://www.alstavista.com>
5. Infoseek, <http://www.infoseek.com>
6. Lycos, <http://www.lycos.com>
7. Excite, <http://www.excite.com>
8. HotBot, <http://www.hotbot.com>
9. G. Salton, A. Wong, and C. S. Yang, *A Vector Space Model for Automatic Indexing*. <http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstr1.cornell/TR74-218,1974>
10. C. Jenkins, M. Jackson, P. Burden, J. Wallis, *Automatic RDF Metadata Generation for Resource Discovery*, Computer Networks, Volume 31 1305-1320, 1999
11. R. A. Fairthorne, *The mathematics of classification: Towards Information Retrieval*, Butterworths, 1961
12. R. Thompson, K. Shafer, D. Vazine-Goetz, *Evaluating Dewey Concepts as a Knowledge Base for Automatic Subject Assignment*, OnLine Computer Library Centre (OCLC), http://orc.rsch.oclc.org:6109/eval_dc.html, 1997
13. OCLC Forest Press, *Dewey Decimal System Home Page*, <http://www.oclc.org/oclc/fp/index.htm> February 2000
14. T. Koch and A. Ardö, *Automatic classification. DESIRE II D3.6a, Overview of results*, NetLab, Lund Univ. Library Development Dept., <http://www.lub.lu.se/desire/DESIRE36a-overview.html>, February 2000
15. C. Jenkins, M. Jackson, P. Burden, J. Wallis, *Automatic Classification of Web Resources using Java and Dewey Decimal Classification*, Computer Networks and ISDN Systems, Volume 30 646-648, 1998
16. A. Powell, *Resource Organisation And Discovery in Subject-based services (ROADS)*, <http://www.ukoln.ac.uk/metadata/roads/intro.html>, 2000
17. UKOLN Metadata Group, *The role of classification schemes in Internet resource description and discovery*, Work Package 3 of Telematics for Research project DESIRE (RE 1004), http://www.ukoln.ac.uk/metadata/desire/classification/class_tc.htm, 1997.
18. L. Chan et al. *Dewey Decimal Classification: A Practical Guide*. Forest Press, ISBN 0-910608-55-5, 1996
19. R. C. J. van Rijsbergen, *Information Retrieval: Second Edition*, Chapter 3, <http://www.dcs.glasgow.ac.uk/Keith/Chapter.3/Ch.3.html> Butterworths, ISBN 0-408-10775-8, 1981

Dunsmuir Flats - Gregory Ain - Great Buildings Online	http://www.greatbuildings.com/buildings/Dunsmuir_Flats.html	Architecture/Architects/Masters/Ain_Gregory/	Architecture/Architects/Ain Gregory	3 / 3
Los Angeles Architecture - Gregory Ain	http://www-rcf.usc.edu/~liebert/ain.htm	Architecture/Architects/Masters/Ain_Gregory/	Architecture	1 / 3
Christopher Alexander, Doors2-Page1 - Mediamatic - Doors of Perception Conference -	http://www.mediamatic.nl/Doors/Doors2/Alexander/Alexander-Doors2-E1.html	Architecture/Architects/Masters/Alexander_Christopher/	Architecture/Architects/Christopher Alexander	3 / 3
Christopher Alexander	http://www.math.utsa.edu/sphere/salingar/Chris.text.html	Architecture/Architects/Masters/Alexander_Christopher/	Architecture/Architects/Christopher Alexander	3 / 3
Architectour Japan - Tadao Ando	http://www.rahul.net/arctour/aa056.html	Architecture/Architects/Masters/Ando_Tadao/	Architecture/Architects/Ando Tadao	3 / 3
Tadao Ando	http://www.pritzkerprize.com/andorel.htm	Architecture/Architects/Masters/Ando_Tadao/	Architecture/Architects/Ando Tadao	3 / 3
Luis Barragan - Great Buildings Online	http://www.greatbuildings.com/architects/Luis_Barragan.html	Architecture/Architects/Masters/Barragan_Luis/	Architecture/Architects/Luis Barragan	3 / 3
Luis Barragan Studios	http://www.mcvirtual.com.mx/barragan/	Architecture/Architects/Masters/Barragan_Luis/	Architecture/Architects/Luis Barragan	3 / 3
The Pritzker Architecture Prize	http://www.pritzkerprize.com/	Architecture/Awards/	Architecture/Awards/	2 / 2
Rudy Bruner Award - Urban Excellence	http://www.brunerfoundation.org/	Architecture/Awards/	Not classified	0 / 2
Frank Lloyd Wright: Designs for an American Landscape	http://lcweb.loc.gov/exhibits/flw/flw.html	Architecture/Exhibits/	Architecture/Buildings/Residential	-1 / 2
The Danish Wave	http://www.danishwave.dk/	Architecture/Exhibits/	Architecture/Exhibits	2 / 2
The Barn Journal	http://museum.cl.msu.edu/barn/	Architecture/Buildings_and_Structures/Barns/	Architecture/Buildings/Barns	3 / 3
Barns of Underhill	http://www.uvm.edu/~hag/underhill/	Architecture/Buildings_and_Structures/Barns/	Architecture/Buildings/Barns	3 / 3
Washington Lighthouses	http://members.tripod.com/lighthouse7/WALIGHTS.HTML	Architecture/Buildings_and_Structures/Lighthouses/	Architecture/Buildings/Lighthouses	3 / 3
Virginia Lighthouses	http://members.aol.com/valights/virginia.htm	Architecture/Buildings_and_Structures/Lighthouses/	Architecture/Buildings/Lighthouses	3 / 3
Historic Irish Castles	http://www.historic.irishcastles.com/	Architecture/Buildings_and_Structures/Castles/	Architecture/Buildings/Residential	1 / 3
The Castles of Latvia	http://www.sveiks.com/pils/ainavas.html	Architecture/Buildings_and_Structures/Castles/	Architecture/Buildings/Castles	3 / 3
The former Baha'i house of worship Ashkhabad	http://www.BCCA.Org/~cvoogt/Ashkhabad/	Architecture/Buildings_and_Structures/Religious	Architecture/Buildings	2 / 3
Church Crawler	http://members.aol.com/PMDrape10/homepage.htm	Architecture/Buildings_and_Structures/Religious	Architecture/Buildings/Religious	3 / 3
Althorpe	http://www.althorp-house.co.uk/	Architecture/Buildings_and_Structures/Residential/	Not classified	0 / 3
Harewood House	http://www.harewood.org/shocked_index.htm	Architecture/Buildings_and_Structures/Residential/	Architecture/Buildings/Residential	3 / 3
The CN Tower	http://www.cntower.ca/	Architecture/Buildings_and_Structures/Skyscrapers/	Not classified	0 / 3
The High Rise Site	http://www.high-rises.co.uk/	Architecture/Buildings_and_Structures/Skyscrapers/	Architecture/Buildings/Skyscrapers	3 / 3
Daniel's Mill Watermill Bridgnorth Shropshire UK	http://www.stargate-uk.co.uk/adverts/daniels.html	Architecture/Buildings_and_Structures/Water_Mills/	Not classified	0 / 3
Haxted Watermill Museum	http://www.mistral.co.uk/hammerwood/haxtmill.htm	Architecture/Buildings_and_Structures/Water_Mills/	Architecture/Buildings/Water Mills	3 / 3
Total Score:				61 / 80
Percentage :				76.2%