Journal of Universal Computer Science, vol. 4, no. 9 (1998), 719-736
submitted: 1/9/98, accepted: 10/9/98, appeared: 28/9/98 *© Springer Pub. Co.*

Categorisation by Context[1]

Giuseppe Attardi
Dipartimento di Informatica, Università di Pisa, Italy
attardi@di.unipi.it

Sergio Di Marco
Dipartimento di Informatica, Università di Pisa, Italy
dimarco@di.unipi.it

Davide Salvi
Dipartimento di Informatica, Università di Pisa, Italy
salvi@di.unipi.it

Abstract: Assistance in retrieving of documents on the World Wide Web is provided either by search engines, through keyword based queries, or by catalogues, which organise documents into hierarchical collections. Maintaining catalogues manually is becoming increasingly difficult due to the sheer amount of material on the Web, and therefore it will be soon necessary to resort to techniques for automatic classification of documents. Classification is traditionally performed by extracting information for indexing a document from the document itself. The paper describes the technique of categorisation by context, which exploits the context perceivable from the structure of HTML documents to extract useful information for classifying the documents they refer to. We present the results of experiments with a preliminary implementation of the technique.

Key Words: information retrieval, Web search, text categorisation, hypertext navigation

Categories: H.3.1, H.3.3, H.3.5, H.5.1, I.2.7, I.5.3

1 Introduction

Most Web search engines (e.g. Altavista™ [Altavista], HotBot™ [HotBot], Excite™ [Excite]) perform search based on the content of documents and provide results as a linear list of such documents, typically ranked in order of relevance. The often unsatisfactory aspect of this approach is that the list can be quite long, with many replications, and without any indication of possible grouping of related material. For instance, issuing a query with the keyword „garbage“, one would obtain a list of documents that discuss ecological issues interspersed with documents about garbage collection in programming languages. Splitting the list of retrieved documents into thematic categories would significantly facilitate selecting those documents of more interest to the user.

Notable exceptions to this approach are Lycos™ [Lycos] and Yahoo™ [Yahoo], which maintain a categorisation of part of their search material. Actually Yahoo gave

---

[1] This is an extended version of a paper presented at the WebNet 98 conference in Orlando, Florida. The paper has received a „Top Full Paper Award“.

up its general search service in favour of Altavista [Altavista] and supports only searches within its own catalogue. This allows a more focused search restricted to the documents within a given category and also the results of a query are presented arranged within subcategories.

However both Lycos and Yahoo are based on manual categorisation of documents performed by a small set of well-trained categorisation technicians (even though Lycos™ recently announced the development of an automatic classifier).

It is questionable whether manual classification will be able to scale well with the growth of the Web, which will reportedly reach over 30 terabytes within 2 years, a size larger than the whole US Library of Congress.

First, manual classification is slow and expensive, since it relies on skilled manpower.

Second, the consistency of categorisation is hard to maintain when different human classifiers are involved. Categorisation is quite a subjective task, as other content related tasks like document indexing and hypertext authoring. An experimental study [Cleverdon 84] on manual indexing for Boolean information retrieval systems has shown that the degree of overlap in the keywords selected by two similarly trained people to represent the same document is, on average, no higher than 30%. Similarly, studies on the task of hypertext authoring (i.e. adding links to relevant, or content-related, documents) have found a very small degree of agreement among different human linkers [Ellis 94]. Automatic (or semi-automatic) hypertext authoring has proven to be a better alternative [Salton 94].

Finally, the task of defining the categories to use (hereafter called *catalogue*) is also difficult and subjective, and new categories emerge continuously in many domains. For example, documents relating to ActiveX technology are hard to categorise: they might fall within operating systems or within graphics or within object-oriented programming. None of these categorisations would be satisfactory, since each of them would miss to establish close connections with other related technologies, like CORBA, JavaBeans, etc. In this case it seems that a new category is emerging („Software Components“) which is not currently envisaged. In fact, by browsing the Web, we may discover several pages that contain references to documents relating to these subjects: each such page in fact determines a context for these documents. By exploiting these contexts, an agent should be capable of creating the appropriate category and to discriminate between documents falling within different categories.

A similar problem arises in the organisation of personal material, for instance mail and bookmarks [Maarek 96] and [Weiss 96].

In this paper we investigate a novel technique for *automatic* categorisation, which may be dubbed *categorisation by context*, since it exploits the context surrounding a link in an HTML document to extract useful information for categorising the document it refer to. This technique is complementary to the traditional technique of *categorisation by content* [Yang 94, Schütze 95, Ng 97], where information for categorising a document is extracted from the document itself. Such approach may exploit linguistic analysis to determine relevant portions of the text [Fuhr 91] and then exploits probabilistic or statistical analysis to perform feature selection [Yang 97] and to weight selected features. Categorisation by context instead

exploits relevance hints that are directly provided in the structure of the HTML documents which people build on the Web. Combining a large number of such hints, a high degree of accuracy can be achieved.

Another significant advantage of context-based indexing categorisation is that it can be applied to multimedia material, including images, audio and video [Shrihari 95], since it does not depend on the ability to analyse and index by content the documents to be categorised

Furthermore, the mechanism that we will describe can be used to restructure a catalogue: in fact classification is performed with respect to a base catalogue hierarchy. Therefore, supplying a different catalogue hierarchy will produce a new categorisation. This is quite useful in the non-infrequent case when one discovers that a certain catalogue is no longer appropriate. With other techniques, manual or automatic, re-categorisation of a document set according to a different catalogue requires significant effort and one tries to avoid it. The technique of categorisation by context provides an automatic tool for doing it.

In particular, the technique can be applied for creating bridges between catalogues in different languages, since it allows transferring material from the catalogue in one language into that for another language, which usually has a different structure to reflect a different culture.

Categorisation by context leverages on the categorisation activity that users implicitly perform when they place or refer documents on the Web, turning categorisation, form an activity delegated to a restricted number of specialists, into a collaborative effort of a community of users. By restricting the analysis to the documents used by a group of people, one can build a categorisation that is tuned to the need of that group.

2 Related Work

Categorisation by content builds internal representations of the content of the documents to categorise („indexing“) and possibly of the categories themselves („category profile extraction“). It then uses a measure *similarity* of such representations, such as vector space retrieval [Salton 75] or fuzzy retrieval [Bookstein 76], to perform categorisation.

Hypertext links pointing to the document to be categorised have not been used so far for categorisation, although they have been used as clues for searching documents [Chalmers 98], and for measuring the „importance“ of a Web site [Brin 98].

Vistabar [Marais 97] is a desktop assistant for Web browsing that provides a function to retrieve pages that refer to the current document as a mean to find out what people think about a document. Vistabar has also a categorisation tool, which uses the Yahoo™ classification hierarchy. A category profile is precomputed for each category in Yahoo and Vistabar performs a traditional vector space matching on the weighted word frequencies of a document relative to the corpus, exploiting however the hierarchy of categorisation to prune the search. Vistabar also allows sharing categorisation and annotations by a group of users.

Purcell and Shortliffe [Purcell 95] discuss the shortcomings of traditional retrieval systems and describe a context-based technique applied in the medical domain.

WebTagger™ [Keller 97] is a personal bookmarking service that provides both individuals and groups with a customisable means of organising and accessing Web-based information resources. In addition, the service enables users to supply feedback on the utility of these resources relative to their information needs, and provides dynamically updated ranking of resources, based on incremental user feedback.

Automatic categorisation is the approach used by Northern Light [Northern Light] in their new search service, which dynamically organises search results, creating Custom Search Folders™ of documents with similar subjects, sources, or types. Within each folder a new subset of the original result list is produced containing only more focused results.

Contextual information is exploited in ARC [Chakrabarti 98], a system for automatically compiling a list of authoritative Web resources on a topic. ARC considers two kinds of pages: *authority* pages and *hub* pages. An authority page contains a lot of information about a topic. A hub page is one that contains a large number of links to pages containing information about the topic. The algorithm computes iteratively two scores for each page p, a hub score $h(p)$ and an authority score $a(p)$. Each iteration consists of two steps: (1) replace each $a(p)$ by the weighted sum of the $h(p)$ values of pages pointing to p; (2) replace each $h(p)$ by the weighted sum of the $a(p)$ values of pages pointed to by p. A positive numerical weight $w(p, q)$ is assigned to each link (from page p to page q) that increases with the amount of topic-related text in the vicinity of the *href* from p to q. This weight is computed from the number of matches between terms in the topic description and a window of 50 bytes of text of the href. The topic-related text can be considered contextual information that the algorithm propagates through links reinforcing the belief that a page is an authority on a topic.

3 Architecture

People browse through documents and base their decisions to follow a link on its textual description or on its position (in case of image maps or buttons). At least for textual links, in principle a document should provide sufficient information to describe a document it refers to. HTML style guides [Tilton 95] suggest making sure that the text in a link is meaningful, avoiding common mistakes like using adverbs or pronouns: „The source code is here“ or „Click this“. Even if the link itself is not sufficiently descriptive, the surrounding text or other parts of the document normally supply enough descriptive information. If such information is sufficient to decide whether a document is worth reading, we can assume it is also sufficient to categorise a document.

The technique of categorisation by context consists in extracting contextual information about documents by analysing the structure of Web documents that refer to them. The overall architecture of the task of categorisation by context is described

in [Fig. 1], and consists in spidering Web documents, HTML structure analysis, URL categorisation, weight combination and catalogue update.

3.1 Spidering and HTML Structure Analysis

This task starts from list of URLs, retrieves each document, analyses the structure of the document expressed in terms of its HTML tags. For an introduction to HTML we refer to the HTML Primer [HTML].

The tags considered are currently: <TITLE>, <Hn>, <UL>, <DL>, <OL>, <A>. Whenever one of these tags is found, a context phrase is recorded, which consists of the title within a pair <Hn> </Hn>, or the first portion of text after a <UL> or <DL> tag, or the phrase within a <A> tag. When a <A> tag is found containing an URL, an *URL Context Path* (URL: C_1: C_2: … : C_n) is produced, which consists of the sequence of the context strings so far (C_1: C_2: … : C_n) associated to the URL. Therefore C_1 is the text in the anchor of the URL, and the other C_i are the enclosing contexts in nesting order.

In the analysis tags related to layout or emphasis (<EM>, <B>, <CENTER>, <FONT> etc.) are discarded.

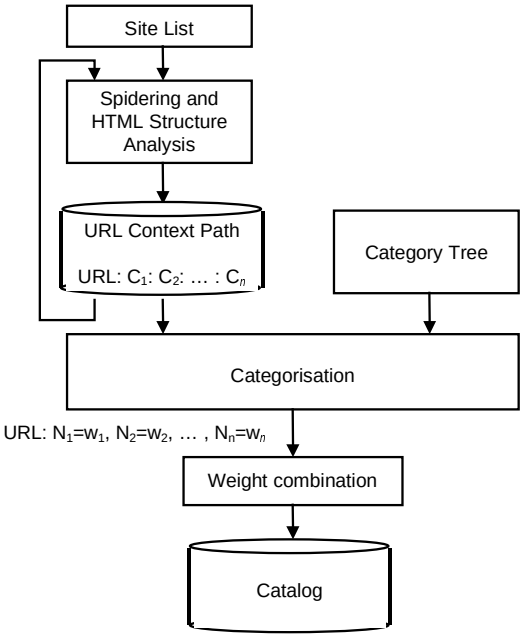

Figure 1: Architecture of Categorisation by Context

Another possible element for a context is the title of a column or row in a table: tag <TH>. Such title can be effectively used as a context for the elements in the corresponding column or row.

For example, consider the following fragment of an HTML page from Yahoo™!:

```
<html>
<head>
<title>Yahoo! - Science:Biology</title>
</head>
<body>
…
<ul>
<li>
<a href="http://esg-www.mit.edu:8001/esgbio/">
M.I.T. Biology Hypertextbook</a> - introductory resource including information on
chemistry, biochemistry, genetics, cell and molecular biology, and immunology.
<li>
<a href="http://muse.bio.cornell.edu/">
Biodiversity and Biological Collections</a>
  - information about specimens in biological collections, taxonomic authority files,
directories of biologists, reports by various standards bodies, and more.
<li>
<a href="http://gc.bcm.tmc.edu:8088/bio/bio_home.html">
Biologist's Control Panel</a> - many biology databases, library and literature links.
<li>
<a href="http://www.molbiol.ox.ac.uk/www/ewan/palette.html">
Biologists Search Palette</a> - a collection of useful search engines for biological
databases on the Internet, accessed through either the Web or gopher.
…
</body>
</html>
```

the following context paths are created:

http://esg-www.mit.edu:8001/esgbio:
„M.I.T. Biology Hypertextbook“ :
„introductory resource including information on chemistry, biochemistry, genetics, cell and molecular biology, and immunology“ :
„Yahoo! - Science:Biology“

http://muse.bio.cornell.edu:
„Biodiversity and Biological Collections“
„information about specimens in biological collections, taxonomic authority files, directories of biologists, reports by various standards bodies, and more“
„Yahoo! - Science:Biology“ :

"http://gc.bcm.tmc.edu:8088/bio/bio_home.html"
„Biologist's Control Panel“
„many biology databases, library and literature links“
„Yahoo! - Science:Biology“ :

"http://www.molbiol.ox.ac.uk/www/ewan/palette.html"
„Biologists Search Palette“
„a collection of useful search engines for biological databases on the Internet, accessed through either the Web or gopher“
„Yahoo! - Science:Biology“ :

Any URL found during the analysis is passed back to the spidering process, if it points to a document within the current site and stored for later analysis if it points to an external site. This allows us to perform a depth-first visit of a site, collecting any categorisation information it contains about itself and other sites.

3.2 Categorisation

The categorisation task exploits the database of URL *Context Path* and the *Category Tree* within which the URL must be categorised. The Category Tree consists of a tree (or a DAG), where each node contains a *title*, i.e. a single word or phrase, which identifies the category.

The goal of the categorisation is to find the most appropriate categories under which an URL should be categorised. The output of the categorisation is a sequence of weights associated to each node in the Category Tree:

URL: $N_1=w_1$, $N_2=w_2$, … , $N_n=w_n$

Each weight w_i represents a degree of confidence that the URL should belong to the category represented by node N_i.

The weights from the Context Path for a URL are added with all other Context Paths for the same URL and normalised. If the weight for a node is greater than a certain threshold, the URL is categorised under that node.

The mechanism should allow for categorising an URL under more than one node, but never in two nodes which are descendant of one another.

4 Algorithm

The categorisation algorithm considers each node in the Category Tree as a path. For instance, the following part of the Arianna category tree (Arianna [Arianna] is a search engine for the Italian Web space that we are using for our experiments):

- Affari e Finanza
  - Assicurazioni
  - Associazioni
  - Banche ed Istituzioni Finanziarie
  - Eventi e Fiere
  - Informazioni e Servizi
  - Pagine delle Aziende
  - Pubblicazioni
  - Scuole ed Istituti di Formazione
- Computer
  - Distribuzione e Vendita
  - Eventi e Fiere
  - Pubblicazioni e Riviste
  - Scuole, Corsi e Associazioni
  - Software e Hardware
  - Telecomunicazioni
- Sport
  - Eventi
  - Fantatornei
  - Notizie ed informazioni
  - Organizzazioni e Società
  - Sport Individuali e di Squadra

corresponds to the following paths, translated into English:

```
Business and Finance: Insurance
Business and Finance: Associations
Business and Finance: Banks
Business and Finance: Events
Business and Finance: Fairs
Business and Finance: News
Business and Finance: Services
Business and Finance: Companies
Business and Finance: Publications
Business and Finance: Schools

Computer: Sale
Computer: Distribution
Computer: Events
Computer: Fairs
Computer: Schools
Computer: Courses
Computer: Associations
Computer: Software
Computer: Hardware
Computer: Telecommunications

Sport: Events
Sport: Fantatornei
Sport: News
Sport: Companies
Sport: Individual
Sport: Team
```

Notice that the category Events appears in several categories, therefore the title of the category is not sufficient to identify the category: the whole path is necessary to disambiguate among the categories. Notice also that we split into two paths any title which contains two terms joined by „e", which literally means „and" but corresponds to the union of the categories, rather than their intersection.

The categorisation algorithm works as follows: for categorising an URL it computes first a vector of matching weights for each path in the Category Tree, then it determines the paths with the best matching vectors, and finally updates the catalogue.

4.1 Computing path match vectors

Given an URL context path (URL: C_1: C_2: … : C_n), the algorithm considers in turn each C_i, starting from C_1. To each level we associate a weight d_l, decreasing from $C_1 = 1$, for instance with a value $1/log_2(n - 1)$. It may be worthwhile to adapt these weights take into account the significance of a tag, for instance a <TITLE> tag may have a slightly higher weight than its position would imply.

Then we extract from C_l the noun phrases it contains n_0, … , n_k, as described in [Section 5.2].

For each path p in the Category Tree, we create a path match vector pv with as many fields as the path length, initialised to 0.

Each phrase n_i is matched against each title in each path. If there is a match between n_i and the title of category in the k-th position in path p, with matching weight mw, then $d_l \times mw$ is added to pv_k. This process is repeated for each level $1 < l < n$.

Notice that since there can be several matching, the value in a field of a path match vector can be greater than one; therefore these values are normalised before performing comparisons.

4.2 Selecting best matching categories

Any path match vector pv that contains non-zero fields is considered as a *potential candidate*. The selection among these candidates is performed as follows:

1. discard any path with 0's in its leading fields. This means that that we found matches only in some subcategory but not in the top-level categories. For instance an URL relating to a business event we matched Events, but not Sports. The URL in fact will match both categories Business and Event in the Business:Event path.
2. among candidates with similar overall score, select the one with longer path. This forces categorisation under the most specific category.

The selected estimate records are stored in the database, associated to the URL. When the same URL is reached form a different path, the new estimates are combined to the previous ones. This will either enforce the indication of the category for a URL or suggest alternative categories for the URL.

5 Implementation

In order to compute the match between a noun phrase in a context and a title in the catalogue, we exploit several tools and data structures.

First, since it is likely that noun phrases do not use the exact terms present in the titles, we must widen somewhat the search. In order to do this we precompute a neighbourhood of words for each term in a title, exploiting information from WordNet, which we store in a neighbourhood table. Second, matching a noun phrase with a title, which in general is also a noun phrase, requires matching multiple words.

5.1 Neighborhood Table

In the following we will denote with **D** the set of titles of categories in the Categorisation Tree, and with **DS** the set of single words in **D**.

For each term t in **DS** we build the set $\mathbf{I}(t)$ of terms in the neighbourhood of t (synonyms, hyponyms, etc.). Moreover, for each term s in $\mathbf{I}(t)$ we denote with $w(s, t)$ the weight of s with respect to t, which depends on whether s is a synonym, hyponym,

hypernym or a related term to *t*. If a term belongs to several of these classes, $w(s, t)$ will take the largest of such weights.

We implement the inverted list of **I** as a hash table **TI** whose keys are elements in **DS**. Since we need to know the matching weight of each pair of words, it is convenient to store in this table not just the matching word but also the matching weight. Therefore the table associates to each word *s* in the neighbourhood of some term *t*, a set of pairs $\langle t, w(s, t)\rangle$, where $t \in$ **DS**.

Table **TI** can be built as follows:

1. $\forall t \in$ **DS**, **TI**$(t) = \{\langle t, 1\rangle\}$
2. $\forall t \in$ **DS**, $\forall s \in$ **I**(t), **TI**(s) = **TI**$(s) \cup \{\langle t, w(s,t)\rangle\}$

Example:

Let '**sport event**' be the title of one category, then *sport event* is in **D**, *sport* and *event* are in **DS**. Table **I** might look like this:

word	neighbourhood
sport	{*football*, *basketball*, *tennis*, … }
event	{*happening*, *match*, …}
…	…

While table **TI** could be like this:

word	related title words
sport	{‹*sport*, 1.0›, … }
event	{‹*event*, 1.0›, … }
football	{‹*sport*, 0.9›, … }
basketball	{‹*sport*, 0.9›, … }
tennis	{‹*sport*, 0.9›, … }
happening	{‹*event*, 0.6›, … }
…	…

5.2 Extracting noun phrases from contexts

Before matching a context with category titles, we extract from the context the noun phrases it contains. For this task we currently use LTPOS [Mikheev 98], a lexicon-based part of speech (POS) tagger with a stochastic disambiguator. For example, the sentence

> *The World Wide Web has evolved into an impressive open structure for sharing information*

is segmented by LTPOS into a series of words each of which is assigned a single POS-tag:

> *The_DT World_NP Wide_NP Web_NN has_VBZ evolved_VBN into_IN an_DT impressive_JJ open_JJ structure_NN for_IN sharing_VBG information_NN*

From such series we consider sequences of adjectives and nouns which form noun phrases, obtaining:

World Wide Web
impressive open structure
information

The example shows that tagging avoids the mistake of considering the term *World* in itself, which would lead to inappropriate categories, but rather suggests to consider it as part of a phrase whose main subject is *Web*.

A POS tagger like LTPOS is essential here, since the lexical information provided by WordNet is not sufficient to determine the proper meaning and role of a word in the context of a phrase.

5.3 Matching noun phrases

The next step is matching a noun phrase from a context with the category titles from the catalogue. For instance we might have the noun phrase *football match* which should match with the title *sport event*.

Let $s = s_0 \ldots s_n$ be a noun phrase consisting of a sequence of words s_i. We must find a category title $t = t_0 \ldots t_n$ where each $s_i \in \mathbf{I}(t_i)$ and to retrieve the corresponding weights $w(s_i, t_i)$.

We can construct all possible sequences:

$t_0 \ldots t_n$

exploiting the information in table **TI**, substituting each s_i with any t_i such that $\langle t_i, w(s_i, t_i)\rangle \in \mathbf{TI}(s_i)$. In order to determine quickly whether a sequence corresponds to a title and in which position in the path the title appears, we use a hash table which associates noun phrases to pairs of a path match vector and the corresponding index. There can be more than one pair since the same title may appear in more than one place in the category tree.

If no match is found, the process is repeated with $s = s_1 \ldots s_n$ and so on. The reason for this is that a noun phrase may be over specific, for instance because of the presence of too many adjectives. By dropping such adjectives, we generalise the phrase and try to match it again.

Having found the weights $w(s_i, t_i)$ for a phrase, there are several ways to combine them to form the overall weight of the match between the noun phrase and the category title. We chose to use the average of $w(s_i, t_i)$ for all i. Such value mw is added to the weight vector of each path which contains the title, as described in [Section 4.1].

6 Experimentation

A prototype tool for categorisation by context has been built in order to verify the validity of the method.

An HTML structure analyser has been built in Perl, derived from the analyser used in Harvest.

A spidering program has been written in Java™, which uses the HTML analyser to produce a temporary file of URL Context Paths.

Also in Java™, we developed a categoriser program that interfaces to WordNet [Miller 95] to perform morphing of the words appearing in the context paths and other linguistic analysis.

We have used the Arianna [Arianna] catalogue for the experiment, translating into English their names, and we tried to categorise a portion of Yahoo™ [Yahoo].

We show the results of categorisation for the first few items present in the page http://www.yahoo.com/Science/Biology:

- MIT Biology Hypertextbook - introductory resource including information on chemistry, biochemistry, genetics, cell and molecular biology, and immunology.
- Biodiversity and Biological Collections - information about specimens in biological collections, taxonomic authority files, directories of biologists, reports by various standards bodies, and more.
- Biologist's Control Panel - many biology databases, library and literature links.
- Biologists Search Palette - a collection of useful search engines for biological databases on the Internet, accessed through either the Web or gopher.

Here are candidate paths for the first URL http://esg-www.mit.edu:8001/esgbio. On the left we display the path and on the right the corresponding match weight vector, which has an entry for each element of the path representing the weight of the match for the corresponding title.

```
science  ....................................  1.0
science : general  ..........................  1.0       0.0
science : earth  ............................  1.0       0.0
science : environment  ......................  1.0       0.0
science : psychology  .......................  1.0       0.0
science : mathematics  ......................  1.0       0.0
science : engineering  ......................  1.0       0.0
science : physics  ..........................  1.0       0.0
science : chemistry  ........................  1.0       0.0
science : space  ............................  1.0       0.0
science : astronomy  ........................  1.0       0.0
science : computer  .........................  1.0       0.0
science : biology  ..........................  1.0       2.13093
science : botany  ...........................  1.0       0.0
technology : biology  .......................  0.0       2.13093
science : archaeology  ......................  1.0       0.0
science : agriculture  ......................  1.0       0.0
```

The grading of candidates for http://esg-www.mit.edu:8001/esgbio produces the following:

```
science  ....................................  1.0
science : general  ..........................  0.5
science : earth  ............................  0.5
science : environment  ......................  0.5
science : psychology  .......................  0.5
science : mathematics  ......................  0.5
science : engineering  ......................  0.5
science : physics  ..........................  0.5
science : chemistry  ........................  0.5
science : space  ............................  0.5
science : astronomy  ........................  0.5
science : computer  .........................  0.5
science : biology  ..........................  1.565465
science : botany  ...........................  0.5
technology : biology  .......................  1.065465
science : archaeology  ......................  0.5
science : agriculture  ......................  0.5
```

Here is the result for the categorisation of one URL from page http://www.yahoo.com/Regional/Countries/Italy/Recreation_and_Sports/Sports/Soccer/Clubs_and_Teams.

Here are candidate paths for the URL http://www.dimi.uniud.it/~cdellagi:

```
sport  ......................................  2.0
sport : individual  .........................  2.0       0.0
sport : team  ...............................  2.0       3.0
sport : club  ...............................  2.0       1.0
sport : event  ..............................  2.0       0.0
sport : news  ...............................  2.0       0.0
```

The grading of candidates for http://www.dimi.uniud.it/~cdellagi:

```
sport  ......................................  2.0
sport : individual  .........................  1.0
sport : team  ...............................  2.5
sport : club  ...............................  1.5
sport : event  ..............................  1.0
sport : news  ...............................  1.0
```

One aspect worth commenting is the skewed distribution of the real-valued numbers that appear in these tables: in fact, the gradings of most of the candidate paths tend to concentrate around a few real numbers such as 1.0, 2.0 or the like. This is primarily due to the fact that no statistical analysis (i.e. term weighting based on term frequency within the context) is performed on contexts, since they are typically short text windows and it is unlikely that the same term occurs more than once in the same context. This situation is reminiscent of what happens in term weighting for information retrieval. Document term weights are often computed by a measure (such as $tf \times idf$, where tf is the term frequency and idf is the inverse document frequency) that depends on the frequency of occurrence of the term in the document. Query term weights are computed instead by a measure (such as idf) independent of such factor, exactly because of the small size of queries and the consequently small likelihood of multiple occurrence of the same term within a query [Salton 88].

The results achieved with the current prototype are quite encouraging. In most cases, the prototype was able to categorise each URL in the most appropriate category. The

few exceptions appeared due to limitations of the linguistic tools we are using: e.g. holes in the WordNet concept tree.

As an experiment to determine the quality of the categorisation, we asked the system to categorise a subset of the Yahoo! pages according to the same Yahoo! catalogue. In principle we should have obtained exactly the original categorisation, and this is what we obtained in most cases. In a few cases the algorithm produced an even better categorisation, by placing a document in a more specific subcategory: for instance a journal on microbiology was categorised under the subcategory of microbiology journals rather than on the category biology journals where it appeared originally.

7 Evolving Categorisation

There are two possible cases when one might feel the need to extend the categories in the catalogue:

- when a category grows too much, containing a large number of documents
- when a large number of documents do not find a proper place in the hierarchy

In order to deal with these problems, the Context Paths should be analysed further.

In both cases, the Context Path for documents in a single large category should be searched for terms in the context, which produce partitions of the URLs. Such partitioning terms should be considered as candidates for new categories. Since several partitioning may arise, statistical analysis techniques will be applied to rank the most promising alternatives and present them to the user who will decide which subcategories to add to the catalogue. The techniques proposed by [Maarek 96] could be applied, but to the context path rather than to the content of the documents, in order to produce possible grouping of documents into subcategories.

The concept hierarchy of WordNet could also provide suggestions of possible subcategories.

8 Context-based Retrieval

We propose to extend the Arianna database to retain the Context Paths produced by the categoriser, the estimate records and the categorisation information for each URL. Exploiting the categorisation information it will be possible:

1. to display the results of a query to Arianna grouped by categories, facilitating the selection of the relevant documents
2. to restrict a search for documents within certain categories
3. to ask within which categories a document appear, helping in finding related documents.

9 Conclusions

We described an approach to the automatic categorisation of documents, which exploits contextual information extracted from the HTML structure of Web

documents. The preliminary results of our experiments with a prototype categorisation tool are quite encouraging. We expect that incorporating further linguistic knowledge in the tool and exploiting information from a large number of sources, we can achieve effective and accurate automatic categorisation of Web documents.

References

[Altavista] *AltaVista*, http://altavista.digital.com.

[Arianna] *Arianna*, http://arianna.it.

[Brin 98] Brin, S., Page, L.: „The anatomy of a large-scale hypertextual Web search engine"; *Computer Networks and ISDN Systems*, 30 (1998) 107–117.

[Bookstein 76] Bookstein, A., Cooper, W.S.: „A general mathematical model for information retrieval systems", *Library Quarterly*, 46 (1976) 153–167.

[Chalmers 98] Chalmers, M., Rodden, K., Brodbeck, D.: „The order of things: activity-centred information access"; *Computer Networks and ISDN Systems*, 30 (1998), 359–367.

[Chakrabarti 98] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., Rajagopalan, S.: „Automatic resource list compilation by analyzing hyperlink structure and associated text"; *Proc. 7th International World Wide Web Conference*, (1998). Available: http://www7.conf.au/programme/fullpapers/1898/com1898.html

[Cleverdon 84] Cleverdon, C.: „Optimizing convenient online access to bibliographic databases"; *Information Services and Use*. 4 (1984), 37–47.

[Ellis 94] Ellis, D., Furner-Hines, J. Willett, P.: „On the measurement of inter-linker consistency and retrieval effectiveness in hypertext databases"; *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, Dublin, IE (1994), 51–60.

[Excite] *Excite*, http://excite.com.

[Fuhr 91] Fuhr, N., Buckley, C. A.: „Probabilistic Approach for Document Indexing"; *ACM Transactions on Information Systems*, 9, 3 (1991), 223–248.

[Keller 97] Keller, R.M., Wolfe, S.R., Chen, J.R., Rabinowitz, J.L., Mathe, N.: „A bookmarking service for organizing and sharing URLs"; *Proceedings of the Sixth International WWW Conference*, Santa Clara, CA (1997). Available: http://ic-www.arc.nasa.gov/ic/projects/aim/papers/www6/paper.html

[HotBot] HotBot, http://hotbot.com.

[HTML] „HTML Primer"; NCSA. Available: http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html.

[Lycos] *Lycos*, http://lycos.com

[Mikheev 98] Mikheev, A.: „Part-of-Speech Guessing Rules: Learning and Evaluation"; *Computational Linguistics*, (in press). Available: http://www.ltg.ed.ac.uk/software/pos.

[Maarek 96] Maarek, Y.S., Shaul, I.Z.B.: „Automatically organizing bookmarks per content"; *Computer Networks and ISDN Systems*, 28 (1996), 1321–1333.

[Marais 97] Marais, H., Bharat, K.: „Supporting cooperative and personal surfing with a desktop assistant"; *Proceedings of ACM UIST'97*, ACM, (1997), 129–138.

[Miller 95] Miller, G.A.: „WordNet: a lexical database for English"; *Communications of the ACM*, 38, 11 (1995), 39–41.

[Ng 97] Ng, H.T., Goh, W.B., Low, K.L.: „Feature selection, perceptron learning, and a usability case study for text categorization"; *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, Philadelphia, USA (1997), 67–73.

[Northern Light] *Northern Light*, http://www.northernlight.com.

[Purcell 95] Purcell, G.P., Shortliffe, E.H.: „Contextual models of clinical publications for enhancing retrieval from full-text databases"; Technical report KSL-95-48, Knowledge Systems Laboratory, Palo Alto, USA (1995).

[Salton 75] Salton, G., Wong, A., Yang, C.S.: „A vector space model for automatic indexing"; *Communications of the ACM*, 18 (1975), 613–620.

[Salton 88] Salton, G., Buckley, C.: „Term-weighting approaches in automatic text retrieval"; *Information processing and management*, 24 (1988), 513–523.

[Salton 94] Salton, G., Allan, J., Buckley, C., Singhal,: „A. Automatic analysis, theme generation, and summarization of machine-readable text"; *Science*, 264 (1994), 1421–1426.

[Schütze 95] Schütze, H., Hull, D.A., Pedersen, J.O.: „A comparison of classifiers and document representations for the routing problem"; *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, Seattle, USA (1995), 229–237.

[Shrihari 95] Srihari, R.K.: „Automatic Indexing and Content-Based Retrieval of Captioned Images“; *Computer*, 28, 9 (1995), 49–56.

[Tilton 95] Tilton, E.: „Composing Good HTML“; CMU, Pittsburgh, USA. Available: http://www.cs.cmu.edu/People/tilt/cgh

[Weiss 96] Weiss, R., Velez, B., Sheldon, M.A., Nemprempre, C., Szilagyi, P., Duda, A., Gifford, D.K.: „HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering“; *Proceedings of the Seventh ACM Conference on Hypertext*, Washington, USA (1996).

[Yahoo] *Yahoo!*, http://yahoo.com.

[Yang 94] Yang, Y.: „Expert network: effective and efficient learning from human decisions in text categorisation and retrieval“; *Proceedings of SIGIR-94*, *17th ACM International Conference on Research and Development in Information Retrieval*, Dublin, IE (1994), 13–22.

Acknowledgments

We thank Antonio Converti, Domenico Dato, Antonio Gullì, Luigi Madella, for their support and help with Arianna and other tools. Fabrizio Sebastiani provided constructive criticism.

This work has been partly funded by the European Union, project TELEMATICS LE4-8303 EUROsearch.