Automatic Web Page Categorization by Link and Context Analysis

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

Antonio Gullì
Ideare SRL, Pisa, Italy
gulli@ideare.com

Fabrizio Sebastiani
Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche, Pisa, Italy
fabrizio@iei.pi.cnr.it

Abstract: Assistance in retrieving documents on the World Wide Web is provided either by search engines, through keyword-based queries, or by catalogues, which organize documents into hierarchical collections. Maintaining catalogues manually is becoming increasingly difficult, due to the sheer amount of material on the Web; it is thus becoming necessary to resort to techniques for the automatic classification of documents. Automatic classification is traditionally performed by extracting the information for representing a document ("indexing") from the document itself. The paper describes the novel technique of *categorization by context*, which instead extracts useful information for classifying a document from the context where a URL referring to it appears. We present the results of experimenting with Theseus, a classifier that exploits this technique.

1. Introduction

Assistance in retrieving documents on the Web is provided by two kinds of tools*: search engines* and *classified directories* (also known as *catalogues*).

Search engines allow keyword-based searches on the content of large collections of Web documents. The results of a query are displayed as a linear list of documents, typically ranked in order of estimated degree (or probability) of relevance. Unfortunately the list is often quite long, and users have neither the willingness nor the skills necessary to perform complex Boolean queries to narrow the search. A recent survey shows that the average query contains less than 3 keywords, and that Boolean queries are rarely used [Jansen 98]. Since the Web contains material of quite different varieties, users cannot anticipate what is available, find difficult to express their interests, and get surprising results because of word ambiguities. Classified directories organize a usually smaller subset of Web material into a hierarchy of thematic categories: each category lists Web pages (or sites) deemed relevant to that category.

AltaVista™ [AltaVista], Hotbot™ [Hotbot] and Infoseek™ [Infoseek] are among the foremost general search engines, while Inktomi™ [Intomi] and Excite™ [Excite] specialize in providing search technology to search services in restricted domains. Lycos™ [Lycos] and Yahoo!™ [Yahoo!] are instead among the best-known Web directories. A recent trend is visible towards the integration of the two kinds of services: AltaVista™ [AltaVista] now comprises also a catalogue-based service, and Lycos plans to merge its services through its acquisition of HotBot. Northern Light™ [Northern Light] provides an interesting combination in its search service, which dynamically organizes the results of a keyword search into groups with similar subject, source or type.

Users have shown to appreciate catalogues. By navigating in a catalogue and docking to the category *c* of interest, a user may either (a) directly access relevant sites pre-categorized under *c*, or (b) perform keyword-based searches restricted to the documents within *c*. The value of a catalogue can be expressed in terms various parameters. These include the quality of its classification scheme (i.e. how intuitive, complete, well-ordered and concise it is), its authoritativeness (how trustworthy the user considers the catalogue), its accuracy (how appropriate the assignment of a document to a category is), its consistency (whether similar documents are classified in a similar way), its timeliness (how quickly it reflects changes in the

document collection), its completeness (how many among the documents relevant to a given category are actually listed therein), and its selectivity (how relevant to a given category the documents listed therein are). The last two aspects (which are related to the measures of recall and precision, well-known in information retrieval) are conflicting, since sometimes a more selective catalogue, i.e. one choosing to exclude some material of lower quality, may be preferable to more complete listings.

Achieving these qualities in a catalogue is a difficult task. Deciding whether a document *d* should be categorized under category *c* requires, to some extent, an understanding of the meaning of both *d* and *c*. Because of this, categorization has traditionally been accomplished manually, by trained human classifiers. This is clearly unsatisfactory since:

- the categorization of any significant portion of the Web requires too much skilled manpower (Srinija Srinivasan, from Yahoo!, acknowledges that the Web grows too fast for human categorizers to keep pace);
- manual categorization is too slow to keep a catalogue up to date with the evolution of the Web. New documents are published, old ones are either removed or updated, new categories emerge, old ones fade away or take up new meanings. Keeping abreast of this evolution by manual means only is practically impossible;
- manual categorization does not guarantee in itself the quality of the resulting catalogue, since categorization decisions are always highly subjective [Cleverdon 84].

Techniques for the automatic, or semi-automatic, classification of Web pages are starting to be exploited on a large scale[1]: beyond the already mentioned Northern Light, Lycos and Arianna [Attardi 99] have recently made available online automatically or semi-automatically built portions of their catalogue, while Inktomi uses automatic classification in building sites like the Disney Internet Guide.

Automatic classification is typically performed by comparing representations of documents with representations of categories and computing a measure of their similarity. In many cases category "profiles" are built by specialist librarians [Northern Light] combining information from several sources (general classification indexes, specialized thesauri, etc.). Techniques for automatically deriving representations of categories ("category profile extraction") and performing classification have been developed within the area of *text categorization* [Ittner 95, Lewis 96, Ng 97, Schütze 95, Yang 94, Yang 97], a discipline at the crossroads between information retrieval and machine learning. Text categorization uses machine learning techniques to inductively build representations of a given set of categories from a *training set* of documents pre-categorized under them. An automatic process can then compare the representation of these categories with the representation of a given document *d* in order to decide to which of these categories it belongs. Alternatively, a document can be compared to previously classified documents and placed in the categories where its most similar documents have also been placed [Yang 94], thus avoiding the need for the construction of explicit category profiles.

Document representations are typically obtained through standard *indexing* techniques developed within the field of information retrieval [Salton 88]; measures of similarity, such as the ones embodied in vector space retrieval [Salton 75] or fuzzy retrieval [Tahani 76], are then computed in order to perform actual categorization. Alternatively, approaches based on the

[1] In this paper, by *automatic classification* we will mean the automatic construction of automatic classifiers, i.e. programs which can automatically classify documents; by *semi-automatic classification* we will mean the manual construction of such automatic classifiers.

identification of concepts have also been proposed which exploit neural network techniques [Autonomy] or linguistic and semantic analysis [InQuizit].

All the approaches to categorization mentioned so far perform what we might call *categorization by content*, since information for categorizing a document is extracted from the text of the document. Categorization by content does not exploit an essential aspect of a hypertext environment like the Web, namely the structure of documents and the link topology. In this paper we investigate a novel technique for automatic categorization, which we have dubbed *categorization by context*, since it exploits the context surrounding a link in an HTML document to extract useful information for categorizing the document referred to by the link. Categorization by context exploits relevance hints that are present in the structure and topology of the HTML documents published on the Web. Combining a large number of such hints, an adequate degree of accuracy of classification can be achieved.

Categorization by context has the significant advantage that it can deal also with multimedia material, including images, audio and video [Srihari 95, Guglielmo 96, Harmandas 97], since it does not rely on the ability to analyze the content of the documents to classify. Categorization by context leverages on the categorization activity that users implicitly perform when they place or refer to documents on the Web, turning categorization, from 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 categorization that is tuned to the needs of that group.

In this paper we report on our experience in building Theseus[1] [Teseo], an automatic classifier of Web documents that exploits the context of links in Web documents [Attardi 98].

2. Improving Web search engines

Several approaches have been attempted for improving the services provided by Web search engines.

AltaVista provides a "refine" capability, whereby users receive suggestions about terms to include or exclude for improving the query. The problem with this approach is that suggested terms are only statistically related to query terms and rarely represent useful semantic concepts: the refined query narrows the focus of search, but does not necessarily direct the search towards the topic of interest. Moreover the approach puts additional burden on the user while providing limited benefits.

Infoseek provides grouping of query results, and also allows retrieving pages similar to a given one, thus providing a limited form of relevance feedback [Harman 92]. Related pages take the user to a parallel, possibly overlapping set of classified documents, but not necessarily to more focused ones.

Northern Light has introduced the technique of Custom Search Folders™: results of traditional keyword searches are dynamically organized into folders containing documents with similar subject, source, or type. When a folder is opened, a new subset of the original result list is produced containing more focused results. To implement the service, Northern Light pre-classifies automatically its whole collection of documents according to (a) subject, using a subject hierarchy of 20,000 terms hand-crafted by human specialists, (b) type, in a shallow hierarchy of 150 document types, (c) language, and (d) source (collection, home page, educational site, etc.). By performing the classification in advance, the grouping of query results can be produced quickly and effectively. Custom Search Folders are dynamically

[1] Theseus was the Greek hero who, helped by Ariadne in getting out of the Labyrinth, killed the Minotaur. This name was chosen since the technique has been developed in connection with the Arianna [Arianna] search engine (Arianna is the Italian name of Ariadne).

created, as opposed to the static structure of a manually built catalogue like Yahoo!'s. As an example of the benefits of this approach, consider a query on "citation processing". A typical search engine would return produce a list of several hundreds of documents, whose topics would include driving violation, information retrieval, data processing, etc. With Infoseek, one can select among the results, for instance, a journal on information retrieval and ask for related pages; however this produces a list of 5 times more pages, apparently drifting into documents unrelated to the original query. Northern Light produces instead a number of search folders, including ones on probation, government sites, archive & record management, document management, office equipment, information retrieval; further, within the latter folder, one finds folders about clustering, relevancy ranking, and bubl.ac.uk (the site of the journal Information Processing and Management), that indeed help in discriminating among documents.

Automated categorization techniques may lead to better Web retrieval tools that provide access to large amount of up-to-date documents, like search engines, with the added convenience of selecting among properly organized material, like in classified directories. Lycos is now providing a catalogue built through automatic classification; however the new catalogue consists only of links, with no summary or other useful information regarding the contents of sites. Lycos uses relevance feedback from users and sorts each list of recommended Web sites accordingly. ACAB [Attardi 99] uses semi-automatic classification techniques based on matching hand-crafted category profiles with the contents of documents, and also builds an automatic summary of page contents which is displayed in the catalogue.

Infoseek has developed a tool called CCE, for Content Classification Engine [CCE], which organizes information automatically into categories. CCE can quickly create a basic directory structure with two techniques: 1) it can import and analyze information from a site map and classify documents accordingly; 2) also, by examining the directory structure of a server, it can make educated guesses about how to categorize documents. For example, all documents relating to employee benefits might be kept in a benefits directory on a server, while employee contact details might be in another directory. If this type of structure exists, then CCE will take advantage of it to build a category tree.

3. Categorization by context

Categorization by context is a technique for automatic Web page categorization based on the following hypotheses:

1. a Web page which refers to a document must contain enough hints about its content to induce someone to read it;
2. such hints are sufficient to classify the document referred to.

Indeed, a document would never be visited, except in casual browsing or through a direct referral, unless there were perceivable clues of its possible interest to potential readers. When people browse through documents, they base their decisions to follow a link on its textual description (or on its position, in case of image maps or buttons). At least in the case of 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, thus avoiding common mistakes such as using adverbs or pronouns (as in "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 assume it is also sufficient to categorize this document.

The classification task must then be capable of identifying such hints. One obvious hint is just the *anchor text* of the link (i.e. the text between the <A> and </A> tags). But additional hints may be present elsewhere in a page: the page title, the section titles, list descriptions, etc. Our idea is to exploit the structure of HTML documents to extract such hints. Moreover, a page may have been reached by following a link from some other page, whose context as well may be relevant, although to a lesser extent. Categorization by context thus exploits both the structure of Web documents and Web link topology to determine the context of a link. Such context is then used to classify the document referred to by the link.

We may also think of the "categorisation by *content* vs. categorisation by *context*" opposition under another light: categorizing document *d* by context is equivalent to categorizing by content another document *b(d)* made by juxtaposing all contexts of occurrence of links referring to *d*. We call *b(d)* the *blurb* of *d*. We are familiar with the notion of "blurb" from e.g. blockbuster novels: the blurb of a book *d* is the list of excerpts from (usually favorable) reviews of the book that the publisher prints on the back of the book in order to encourage prospective customers in buying it. Conceptually, the blurb of a book may then be understood as "what the others say about the book". Likewise, while the content of Web document *d* is, in some sense, what the *author* says about *d*, the blurb of *d* is "what the *Web* says about *d*". The advantage of analyzing, instead of *d* itself, the blurb of *d*, is that typically one refers to a document with a more concise description and with more significant terms than those used in the document itself. This simplifies categorization since the presence of terms which mislead the classifier is less likely.

Hypertext links pointing to the document to be categorized have not been used so far for categorization, although they have been used as clues for searching documents [Chalmers 98, Li 98], and for measuring the "importance" of a Web site [Brin 98]. Contextual information is also exploited in ARC [Chakrabarti 98], a system for automatically compiling lists of authoritative Web resources on a topic, which is discussed in detail in Section 8.

4. Architecture

The task of categorization by context consists in extracting contextual information about documents by analyzing the structure of Web documents that refer to them. The overall architecture of the task is described in Figure 1; the subtasks, to be carried out in sequence, are spidering Web documents, HTML structure analysis, URL categorization, weight combination and catalogue update. See the full paper [Attardi 99a] for a detailed description of the adopted algorithm.

4.1 Spidering and HTML Structure Analysis

This task starts from a list of URLs, retrieving the documents referred by each of them and analyzing the structure of the document expressed in terms of its HTML tags (for an introduction to HTML see 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 a URL, a *URL Context Path* (URL: C_1: C_2: … : C_n) is produced, which consists of the sequence of the context strings (C_1: C_2: … : C_n) so far associated to the URL. Therefore C_1 is the text in the anchor of the URL, and the other C_I's are the enclosing contexts in nesting order. In the analysis, tags related to layout or emphasis (<EM>, <B>, <CENTER>, <FONT> etc.) are discarded. Another possible element for a context is

the title of a column or row in a table, i.e. tag <TH>. Such title can be effectively used as a context for the elements in the corresponding column or row.

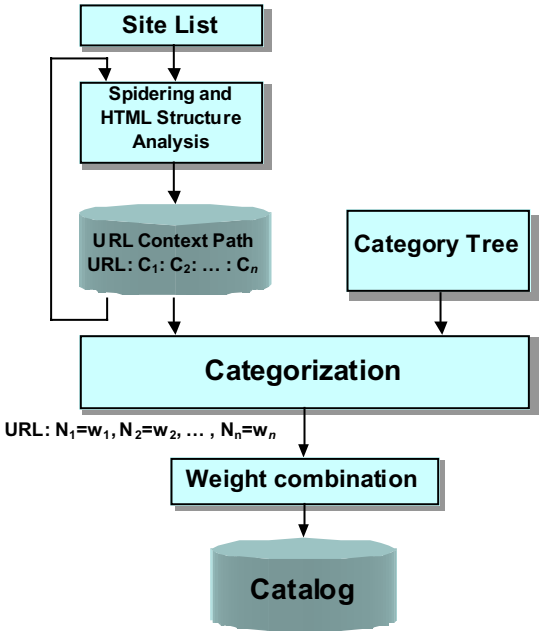

Figure 1: Architecture of Categorization by Context

Throughout the paper we will use the following example, consisting of a fragment of the HTML page `http://www.yahoo.com/Science/Biology` from the Yahoo! catalogue:

Home: 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.

The HTML source for this page is:

```
<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 to perform a depth-first visit of a site, collecting any categorization information it contains about itself and other sites.

4.2 URL Categorization

The categorization task exploits the database of URL *context paths* and the *category tree* within which the URL must be categorized. 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 categorization is to find the most appropriate categories to which a URL should belong. The output of the categorization is a sequence of weights associated to each node in the category tree:

URL: $N_1=w_1, N_2=w_2, \ldots, N_n=w_n$

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 normalized. If the weight for a node is greater than a certain threshold, the URL is categorized under that node. The mechanism allows for categorizing a URL under more than one node, but never in two nodes which are descendant of one another.

5. Theseus

Theseus is a tool for performing categorization by context that has been built in order to verify the validity of the method. Theseus works currently for English and Italian using TreeTagger [Schmid 94], a language-independent part-of-speech tagger, for which English, Italian, French and German lexicons are available.

An HTML structure analyzer has been built in Java™ which builds a context tree for each HTML page by analyzing the parse tree produced by an HTML parser written in Perl. The spidering program is written in Java and uses the HTML analyzer to produce a database of URL context paths. We have developed, also in Java, a categorizer program that interfaces to TreeTagger to perform lexical analysis of the sentences appearing in the context paths. TreeTagger performs part-of-speech tagging and morphing, and returns the lemmas for the words in each sentence.

Spidering and categorization are performed by exploiting a transaction system: each operation is recorded in persistent storage, so that it can be resumed if a failure of any type (e.g. failure in the connection, or interruption in the program) happens. The transactional data base is implemented by interfacing through Java Native Invocation to the GNU gdbm data base system.

We have used the Arianna [Arianna] catalogue for the experiment, and built catalogues from both its English and Italian collections of Web pages.

5.1 Exploiting noun phrases

The benefits of linguistic analysis in information retrieval have always been controversial. In order to determine whether performing noun phrase analysis improves or not the effectiveness of classification, we have compared the classifier with a version that does not perform any analysis of noun phrases extracted from contexts. The following table shows the resulting

overall difference in the number of entries placed in the top-level categories of the two catalogues:

Category	Without	With
scuola e istruzione	955	971
salute	288	311
sport	765	731
informazione e notizia	322	327
organizzazione sociale	1	1
politica e società	2440	2550
arte e cultura	1093	1084
intrattenimento	650	628
viaggio e turismo	695	683
computer, internet e telecomunicazioni	602	576
economia	1518	1462
tempo libero	172	184

In fact most of the differences arise from differences in the analysis of noun phrases. For example, the sentence "interno dell'omonimo liceo" ("inside the high school with the same name") misleads the classifier to place an entry within the category "liceo" ("high school"), and similarly for "studentessa di liceo" ("a high school female student"). In fact both sentences contain the word "liceo" but as an indirect reference. Noun phrase analysis in both cases detects that the subject of the phrase is not "liceo" ("high school") and so it avoids attributing a high weight to the occurrence of such term.

Overall, in this experiment, there has been an improvement of approximately 5% in effectiveness by performing noun phrase analysis.

5.2 Identifying site structure

Quite often the pages of a site have a characteristic structure represented by links across pages. For instance there can be references to the main page, or links to the general services available in the site, like searching within the site, help or information pages. Finally, there can be advertisement banners in precise positions in each page. We want to avoid classifying such pages. In order to identify these structural links, we perform an initial breadth-first analysis of pages reachable from the starting page, currently limited to a depth of 3. Any links which are repeated at a frequency above 90% of the overall number of visited pages are considered structural links, placed in a stop list of URLs and discarded in the subsequent analysis of the site.

5.3 Link identification

Sometimes links are embedded within CGI references, e.g. `HREF="/cgi/go?http://www.inrete.it/classica"`. This technique is used by site administrators to keep track of the links that users select from their pages. This can be useful for several purposes: determining which pages are more interesting to users, measuring the number of page referrals towards other sites (this is particularly important for advertisers' sites), reordering the content of a page by placing the most requested URLs in more prominent position, etc. Identification of such embedded links is performed by Theseus in the initial phase of site analysis. If a repeated pattern containing URLs in the HREF fields is detected, the URLs themselves are stripped from such pattern.

5.4 Site border identification

Site border identification is essential since in Theseus a document to be classified is any URL which lies outside the border of the current site. A first approximation is to consider as external any URL that has a different prefix. However, sometimes this is not sufficient, and we might have to perform iterative analysis of link topology, as in [Chakrabarti 98], in order to identify such borders.

5.5 Integration with a general search engine

There are several benefits in performing classification in close integration with a search engine. The most obvious one is to avoid a separate spidering of Web documents. The spidering performed by the search engine can be used also by the classifier, to which any new document discovered by the spider is passed.

We also plan to exploit the Arianna search engine in the identification of initial sites for the categorization. An HTML page can be considered as a suitable source for categorization if it contains a large number of links to external pages. In the terminology of [Chakrabarti 98] these pages are called *hubs*. From the indexing information maintained by Arianna, it is possible to obtain such information just by querying its database. However it is important to select as initial sites pages whose content is authoritative. Therefore the algorithm of ARC should be used to identify authoritative hubs. The search engine can be exploited also to provide support for queries within categories. It is sufficient to provide the list of documents within each category so that the search engine can index them. Vice-versa, category information produced by the classifier can be used in the search engine to improve the presentation of query results, grouping them by categories, like in Northern Light and ACAB.

6. Assessment

The results achieved with the current prototype are quite encouraging. In most cases, the prototype was able to categorize each URL in the most appropriate category. The few exceptions appeared due to limitations of the linguistic tools we used for building word neighborhoods: e.g. holes in the WordNet concept tree.

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

The performance of Theseus is also satisfactory: it classifies approximately 500 sites per hour. Examples of catalogues built using Theseus are available at http://medialab.di.unipi.it/Project/Arianna/Teseo. The largest one lists over 27000 documents and was built in two runs of approximately 4 hours each. Figure 2 shows the main page of such catalogue. For each category, the number of documents within that category or its subcategories is shown.

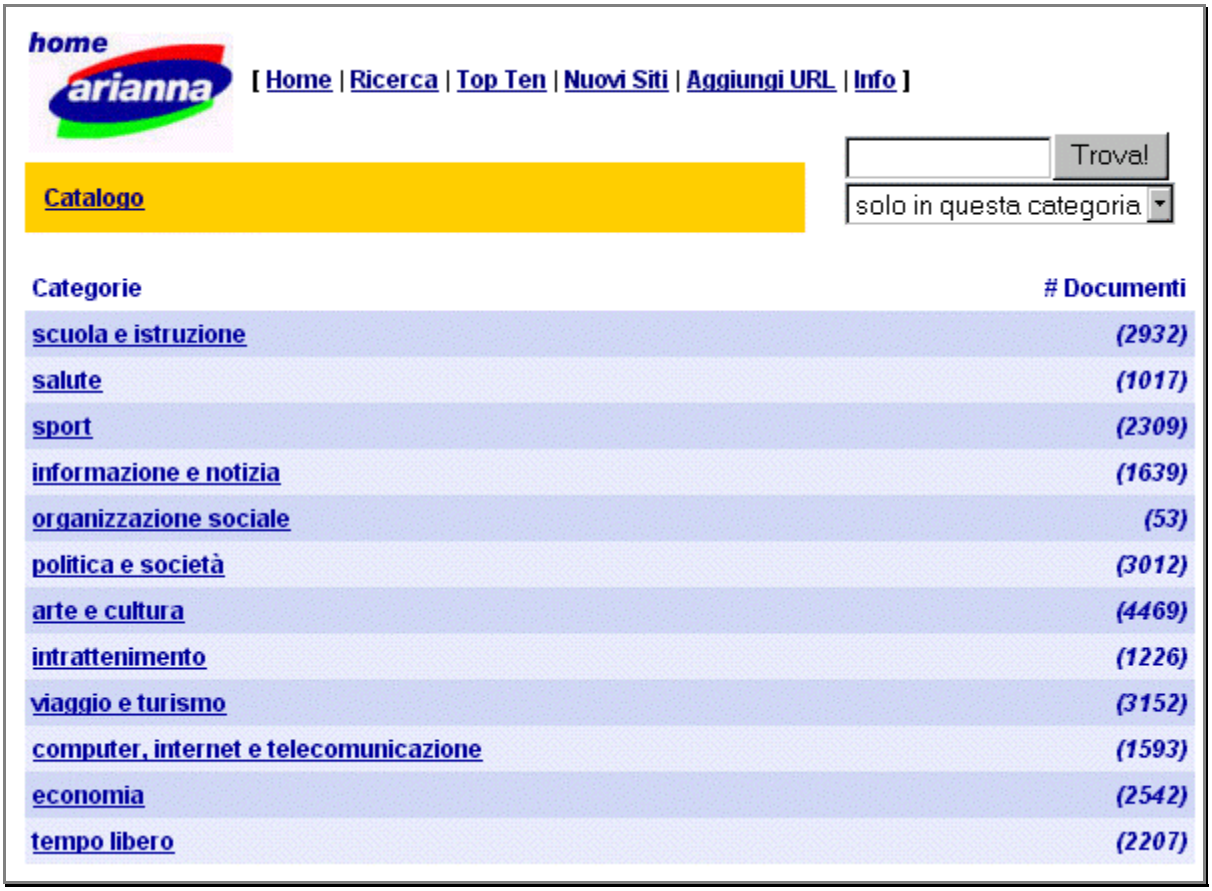

Figure 2. The top page of Theseus's catalogue.

Figure 3 shows a portion of the catalogue page on search engines. Each entry consists of a rank value, a URL, and the closest contexts for the URL (there can be more than one if the document is linked from several pages). The anchor for the URL is also extracted from a context. For purposes of analysis, at the moment we also show the noun phrases that contributed to classifying the document within this category.

We have compared the catalogue built by Theseus with the one built by ACAB using categorization by content on the Italian Web space. For instance Theseus placed 180 documents in the category "Search Engines"; ACAB instead found over 500; however many of these where not pages about search engine, but pages with links to search engines or which mentioned search engines. In this case Theseus appeared to be more precise. Another difference is that Theseus typically detects the main page of a site or of a compound document, since the main page is more likely to be referred by other documents. On the other hand, classification by content cannot easily distinguish between main page and other pages, even though certain heuristics can be applied [Attardi 99].

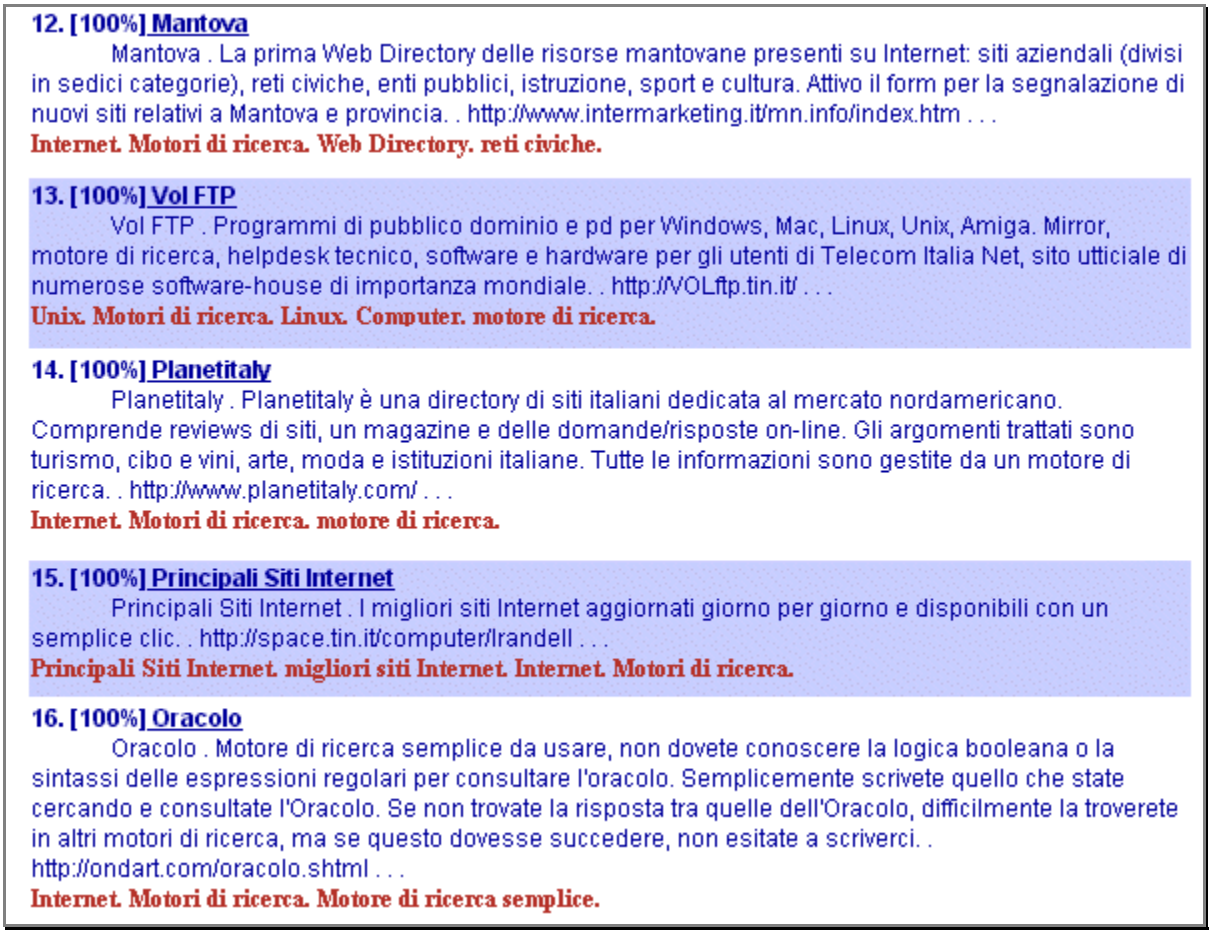

Figure 3. A portion of a catalogue page on Search Engines.

7. Open Issues

The neighborhood table is a critical component of the present implementation. The quality of categorization depends on the quality of neighborhoods.

In our first implementation we relied on WordNet for building such neighborhoods. We extracted for each word in a category title its synonyms, hyponyms and related words. The weights $w(s, t)$ were assigned according to whether t was a synonym, hyponym or a related word with s. We noticed that the use of related words introduces noise, but on the other hand by discarding them completely we miss some important connections, for instance between "military" and "arm". Therefore we had to give low weights to related words.

For the experiments with Italian documents, we could not use WordNet, since the Italian version is still under construction. A first approximation was to use the English version of neighborhoods and to translate them into Italian. This also proved unsatisfactory and we had to revise them significantly by hand. The neighborhood of a word is equivalent to a traditional category profile, when the category title contains a single word. However, when a title contains several words, neighborhoods produce some crosstalk. Therefore we are planning to implement category profiles also for multiple words titles.

In building category profiles we have several options: create them by hand, possibly by means of some interactive tools like in ACAB [Attardi 99], or use learning techniques like those by [Ittner 95, Lewis 96]. The latter techniques requires a training set of categorized documents, so it raises problems of bootstrapping. A possible solution is to start with a catalogue built with Theseus with minimal category profiles, made just with synonyms of

titles. A learning phase could then be applied to such catalogue for extending the category profiles.

Another issue is the proper ranking of documents in the catalogue. Our experience shows that if we start from authoritative sites, the algorithm performs fairly well and produces pages in an acceptable ranking order. The problem arises when we let the classifier crawl freely among sites. We should value differently the contribution to categorization from different sites: we could use the authoritative measure of ARC or the page ranking schema of [Page 98].

8. Related work

Citation processing is any retrieval technique in which documentary citations are traced to identify documents related to a given one. Citation processing techniques are typically used for quantitative analysis, for instance to measure the "impact factor" of scientific journals. Given two documents d_1 and d_2, incoming citations are used in computing *co-citation strength*, i.e. the number of documents that quote both d_1 and d_2. Outgoing citations are used to compute the *bibliographic coupling* [Kessler 63], i.e. the number of documents that are quoted by both d_1 and d_2. The Web provides a new opportunity for citation processing to exploit "citation in context so that quantitative data can be augmented with qualitative statements about the work being cited" [Garfield 97]; this is a direct offspring of the application of citation processing to hypertext information systems, where hypertext links are interpreted as citations [Savoy 97]. This suggests a possible evolution of the techniques of citation processing towards more sophisticated techniques of context link analysis, which exploit not only link topology but also the semantic structure of documents. Categorization by context is therefore an early application of context link analysis. Some other Web-based systems, which we review in the following sections, are based on related principles.

The ARC system [Chakrabarti 98] performs automatic resource compilation by using citation processing and limited context analysis. Given a topic t, ARC computes for each Web page p two scores: its *authority value* $a(p)$, which measures how "authoritative" is p with respect to t in terms of the number of t-related pages pointing to it, and its *hub value* $h(p)$, which measures the "informative content" of p on t in terms of the number of t-related pages it points to. The purpose of ARC is to identify, given a topic t, the k most authoritative and k most informative Web documents. ARC is "kick-started" by issuing query t to AltaVista, which provides an initial set of documents relevant to t; an expansion phase follows, in which documents with distance $d \leq 2$ in the citation graph are added to the set. The algorithm computes the scores for each page p iteratively. 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 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 HTML link 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. The algorithm formalizes the intuition that the hub value of a page is high if the page points to many authoritative pages, and that the authority value of a page is high if many informative pages refer the page. The authors report that, for $k = 15$, normally the algorithm *near-converges* (i.e. the identity of the top 15 authority and hub pages stabilizes) in approximately 5 iterations.

Google [Brin 98] is a search engine, which rates Web pages along a single "authoritativeness" scale. The value $P(d)$ of a given page d is iteratively calculated, until near-convergence. Google also avoids spamming by ensuring that a document achieves a highly

rank after a query only if it is quoted by many already highly ranked documents. Google uses several cues other than links for better ranking a document *d*, such as anchor text, the words that appear in the title and in the text of *d*, etc.

[Harmandas 97] describes a technique for searching images on the Web based on analyzing the text of documents that point to documents containing images.

Rankdex [Li 98] is a search engine based on standard information retrieval techniques applied to the anchor text of Web links.

Table 1 summarizes the main features and techniques used in the system we have discussed.

	Theseus [Attardi 98]	**ARC** [Chakrabarti98]	**Google** [Brin 98]	**Rankdex** [Li 98]
Stemming	yes	no	no	no
Stop word removal	yes	no	optional	no
Incoming links	yes	yes	yes	yes
Outgoing links	no	yes	no	no
Statistical weighting	yes	no	no	yes
Syntactic analysis	yes	no	no	no
Lexical resources	yes	no	no	no

Table 1. Comparison chart of automatic Web classifiers

8.1 Parasiting or not parasiting

The algorithm used by ARC systematically removes from its list of authoritative and (especially) informative pages, those pages belonging to super-hubs, i.e. catalogue-based search engines [Chakrabarti 98a]. The rationale for this is to avoid "parasitical behavior", i.e. to avoid exploiting the contribution of those systems ARC intends to compete with. Although on the surface this looks like safe scientific practice, this decision seems debatable, in that (a) the same scientific practice might as well prevent "parasiting" on AltaVista, and, even more importantly, (b) it dwells on the problematic distinction between a hub and a super-hub. More importantly, this decision seems to miss the very point of using context link analysis. These techniques are *parasitical by nature*, in that they purport to determine the relevance, or authoritativeness, of document *d* based not on their own judgment, but on the judgment of others, i.e. based on "what the Web says" of *d*. This of course involves using what hubs and super-hubs say of *d*. Both Theseus and Google do not avoid using super-hubs, and might thus be characterized as parasitical.

9. Conclusions

We described an approach to the automatic categorization of documents, which exploits contextual information extracted from an analysis of the HTML structure of Web documents as well as the topology of the Web. The results of our experiments with a prototype categorization tool are quite encouraging. By exploiting information from several sources, the tool achieves an effective and accurate automatic categorization of Web documents. The tool may evolve by incorporating further linguistic knowledge and techniques for learning category profiles.

Automatic categorization is a complex task and categorization by context is a useful technique that complements the traditional techniques based on content of documents.

10. References

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

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

[Attardi 98] Attardi, G., Di Marco, S., Salvi, D.: "Categorization by context", *Journal of Universal Computer Science*, 4(9):719–736, 1998. Available as http://www.iicm.edu/jucs_4_9/categorisation_by_context.

[Attardi 99] Attardi, G., Gullì, A.: "Towards automated categorization and abstracting of Web sites". Submitted for publication.

[Attardi 99a] Attardi, G., Gullì, A., Sebastiani, F.: "Automatic Web Page Categorization by Link and Context Analysis". Manuscript.

[Autonomy] Autonomy, http://www.autonomy.com.

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

[CCE] Infoseek Content Classification Engine. Available as http://software.infoseek.com/products/cce.

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

[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", *Computer Networks and ISDN Systems*, 30, 65–74, 1998. Available as http://www7.conf.au/programme/fullpapers/1898/com1898.html

[Chakrabarti 98a] Chakrabarti, S. 1998. Personal communication.

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

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

[Guglielmo 96] Guglielmo, E.J., Rowe, N.: "Natural-language retrieval of images based on descriptive captions", *ACM Transactions on Information Systems*, 14(3), 237–267, 1996.

[Harman 92] Harman, D.: "Relevance feedback and other query modification techniques", in Frakes, W. B. and Baeza-Yates, R. (eds.), "Information retrieval: data structures and algorithms", Prentice Hall, Englewood Cliffs, US, 241-263, 1992.

[Harmandas 97] Harmandas, V., Sanderson, M., Dunlop, M.D.: "Image retrieval by hypertext links", *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, Philadelphia, US, 296–303, 1997.

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

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

[InQuizit] InQuizit, http://www.inquizit.com.

[Ittner 95] Ittner, D.D., Lewis, D.D., Ahn, D.: "Text categorization of low quality images", *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, 301–315, 1995.

[Jansen 98] Jansen, B.J., Spink, A., Bateman, J.: "Searches, the subjects they search, and sufficiency: a study of a large sample of Excite searches", *Proceedings of WebNet'98*, Orlando, US, 1998.

[Kessler 63] Kessler, M.M.: "Bibliographic coupling between scientific papers", *American Documentation*, 14, 10–25, 1963.

[Infoseek] Infoseek, http://infoseek.go.com

[Inktomi] Inktomi, http://www.inktomi.com/products/search/

[Lewis 96] Lewis, D.D., Schapire, R.E., Callan, J.P., Papka, R.: "Training algorithms for linear text classifiers", *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, Zürich, CH, 298–306, 1996.

[Li 98] Yanhong Li. Toward a qualitative search engine, *IEEE Internet Computing*, 2(4):24–29, 1998.

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

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

[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, US, 67–73, 1997.

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

[Page 98] Page, L.: "The PageRank citation ranking: bringing order to the Web", *Proceedings of ASIS'98, Annual Meeting of the American Society for Information Science, 1998.*

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

[Salton 88] Salton, G., Buckley, C.: "Term-weighting approaches in automatic text retrieval", *Information Processing and Management*, 24, 513–523, 1988.

[Savoy 97] Savoy, J.: "Citation schemes in hypertext information retrieval". In Agosti, M. and Smeaton, A.F. (eds.), *Information Retrieval and Hypertext*, Kluwer Academic Publishers, Dordrecht, NL, 99–120, 1997.

[Schmid 94] Schmid, G.: "TreeTagger – a language independent part-of-speech tagger". Manuscript, 1994. Available as http://www.ims.uni-stuttgart.de/Tools/DecisionTreeTagger.html.

[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, US, 229–237, 1995.

[Srihari 95] Srihari, R.K.: "Automatic indexing and content-based retrieval of captioned images", *Computer*, 28(9), 49–56, 1995.

[Tahani 76] Tahani, V.: "A fuzzy model of document retrieval systems", *Information Processing and Management*, 12(3), 177-187, 1976.

[Teseo] *Teseo*, http://medialab.di.unipi.it/Project/Arianna/Teseo.

[Tilton 95] Tilton, E.: "Composing good HTML". Manuscript, Carnegie Mellon University, Pittsburgh, US, 1995. Available as http://www.cs.cmu.edu/People/tilt/cgh

[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, 13–22, 1994.

[Yang 97] Yang, Y.: "An evaluation of statistical approaches to text categorization", Technical Report CMU-CS-97-127, School of Computer Science, Carnegie Mellon University, Pittsburgh, US, 1997. Forthcoming on the *Information Retrieval Journal*.

Acknowledgments

We thank Antonio Converti, Domenico Dato and Luigi Madella for their support and help with Arianna and other tools. This work has been partly funded by the European Union under Project TELEMATICS LE4-8303 "EUROsearch".