Theseus: Categorization by Context

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

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

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

1. Introduction

The traditional approach to document categorization is *categorization by content*, since information for categorizing a document is extracted from the document itself.

In a hypertext environment like the Web, the structure of documents and the link topology can be exploited to perform what we call *categorization by context* [Attardi 98]: the context surrounding a link in an HTML document is used for categorizing the document referred by the link.

Categorization by context is capable of dealing also with multimedia material, since it does not rely on the ability to analyze the content of documents.

Categorization by context leverages on the categorization activity implicitly performed when someone places or refers to a document on the Web. By focusing the analysis to the documents used by a group of people, one can build a catalogue tuned to the need of that group.

Categorization by context is based on the following assumptions:

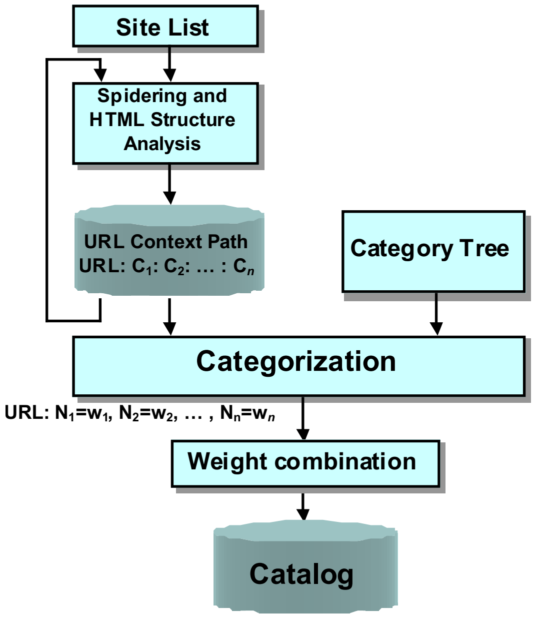
1. a Web page which refers to a document must contain enough hints about its content to suggest reading it
2. such hints are sufficient to classify the document.

The classification task must be capable of identifying such hints. One obvious hint is just the anchor text for the link, but additional hints may be present elsewhere in a page: page title, section headers, list descriptions, etc. All these hints make up the context for the link.

Categorization by context 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 by the link.

2. Architecture

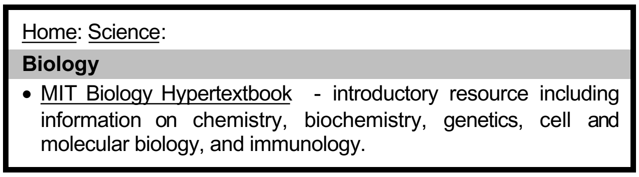
The overall architecture of our system is described in the following figure.


Site List
Spidering and HTML Structure Analysis
URL Context Path
URL: C_1: C_2: ... : C_n
Category Tree
Categorization
URL: N_1=w_1, N_2=w_2, ... , N_n=w_n
Weight combination
Catalog

2.1 Spidering and HTML Structure Analysis

This task starts from a list of URLs, retrieves each document and analyzes its HTML structure. Whenever a structural HTML tag is found, a context phrase is recorded, e.g. the title within a pair <Hn> </Hn>, or the first portion of text after a <UL> or <DL> tag. When a <A> tag is found, 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.

From the following fragment of a Yahoo™! page


Home: Science:

Biology

- MIT Biology Hypertextbook - introductory resource including information on chemistry, biochemistry, genetics, cell and molecular biology, and immunology.

the following context path is produced:

```
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"
```

Any external URL found during the analysis is passed back to the spidering process.

2.2 Categorization

The categorization task uses the database of URL *Context Path* and the *Category Tree*. Each node in the Category

Tree contains a *title*, i.e. a single word or phrase, which describes the category.

The task produces 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 the affinity of the URL document to the category represented by node N_i. The weights from the Context Path for a URL are added with those from other Context Paths for the same URL and normalized.

3. Algorithm

Each node in the Category Tree is represented as a path: e.g. the category Events under Computer is represented by the path Computer: Events while the category Events under Sport is represented by Sport: Events. This helps resolving ambiguities: a match with term Event is not sufficient for categorization; all categories in the path must be matched to confirm the specific meaning of the term.

The categorization algorithm computes first a vector of matching weights for each path in the Category Tree, it selects the best matching vectors, and then it computes from these the weight for the corresponding category. Once all URL have been categorized, a catalogue is built by assigning each URL to the best matching categories.

3.1 Computing path match vectors

Given an URL context path (URL: C_1: C_2: … : C_n) and a category T, the algorithm adds the affinity of each C_i to T, decreased inverse logarithmically with i.

The affinity is computed by extracting noun phrases from the context (by means of TreeTagger [Schmid 94]) and computing matches between variants of the title of T. The variants are obtained by replacing words with synonyms or hyponyms, obtained from WordNet [Miller 95].

For the example above, here are some of the non-zero vectors computed. For each path on the left we display the corresponding match weight vector.

```
science  .................  1.0
science : biology  .......  1.0        2.13093
science : botany  ........  1.0        0.0
technology : biology  ....  0.0        2.13093
science : agriculture  ...  1.0        0.0
```

3.2 Selecting best matching categories

Among the path vectors those with leading 0's are discarded and those with the longest path are selected. This allows disambiguation between senses of words in categories and choosing the most specific category.

4. Theseus

Theseus is a tool for categorization by context built to experiment the approach. Examples of catalogues built using Theseus are available at http://medialab.di.unipi.it/Project/Arianna/Teseo. The largest one lists over 27000 documents.

We compared the catalogue built by Theseus with the catalog built by SearchTone [Gullì 98] using categorization by content. For instance Theseus placed 180 documents in the category Search Engines. SearchTone instead found over 500; many of which where not search engine pages, but pages with links to search engines or which mentioned search engines. Moreover classification by content has difficulties in detecting the main page of a compound document, since it often does not contain enough specific content. A similar problem occurs during the training phase for learning classification algorithms.

4.1 Exploiting noun phrases

The benefits of linguistic analysis in information retrieval have always been controversial. We compared the classifier with a version not performing analysis of noun phrases from contexts. The experiment showed that noun phrase analysis lead to an improvement of approximately 5% in precision.

5. Conclusions

Information extracted from contexts can be useful to perform categorization, as in Theseus, but it can serve other purposes, e.g. for authoritativeness ranking as in ARC [Chakrabarti 98] or [Page 98], for relevance ranking [Boyan 96].

Categorization by context is a useful technique that complements the traditional techniques based on the content of documents.

6. References

[Attardi 98] Attardi, G., Di Marco, S., Salvi, D.: "Categorization by context", Journal of Universal Computer Science, 4(9):719–736, (1998).

[Boyan 96] Boyan, J., Freitag, D., Joachims, T.: "A Machine Learning Architecture for Optimizing Web Search Engines", *AAAI Workshop on Internet Based Information Systems*, 1996.

[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 WWW Conference*, 30:65–74 (1998).

[Gullì 98] Gulli, A., Attardi, G.: "Towards Automated Categorization and Abstracting of Web Sites", submitted for publication, (1998).

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

[Page 98] Page, L.: "The PageRank citation ranking: bringing order to the Web", *Ann. Meeting of America Soc. Info. Sci.* (1998).

[Schmid 94] Schmid, G.: "TreeTagger – a language independent part-of-speech tagger", (1994).

Acknowledgments

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