An Evolutionary Approach to Automatic Web Page Categorization and Updating

Vincenzo Loia and Paolo Luongo

Dipartimento di Matematica ed Informatica,
Università di Salerno,
84081 Baronissi (Salerno), Italy
{loia,pluongo}@unisa.it

Abstract. Catalogues play an important role in most of the current Web search engines. The catalogues, which organize documents into hierarchical collections, are maintained manually increasing difficulty and costs due to the incessant growing of the WWW. This problem has stimulated many researches to work on automatic categorization of Web documents. In reality, most of these approaches work well either on special types of documents or on restricted set of documents. This paper presents an evolutionary approach useful to construct automatically the catalogue as well as to perform the classification of a Web document. This functionality relies on a genetic-based fuzzy clustering methodology that applies the clustering on the context of the document, as opposite to content-based clustering that works on the complete document information.

1 Introduction

The World Wide Web (WWW or Web) is a cheap and powerful environment for sharing information among specialized communities. The unexpected widespread use of the WWW, the presence of heterogeneous data sources, the absence of recognized organization models, make difficult, in many cases frustanting, the task of Internet searching. One solution to this problem is to categorize the Web documents according to their topics. This explains why popular engines (Altavista, Netscape and Lycos) changed themselves from crawler-based into a Yahoo!-like directories of web sites. Just to give an example of the difficulty of this task, Yahoo! maintains the largest directory list composed of 1.2 million of terms thanks to the support of thousands of human editors.

Many researches have been involved in the study of automatic categorization. Good results have been reported in case of categorization of specific documents, such as newspapers [7] and patent documents [11]. Infoseek experimented neural network technology, other approaches have used clusters generated in a dynamic mode [13] [8].

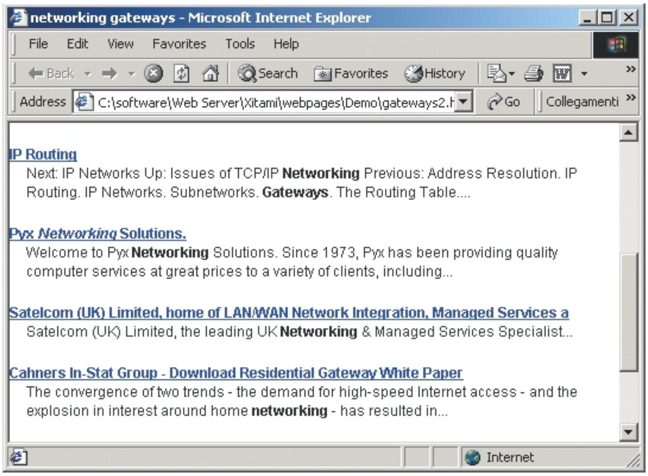
N. Zhong et al. (Eds.): WI 2001, LNAI 2198, pp. 292–302, 2001.

The impressive evolution of the Web makes difficult the management of consistent category directories. This drawback has an immediate effect in a lost of precision reported by the most popular Web search engines (they return only a fraction of the URLs of interest to user [14], have a small coverage of available data [10], suffer of instability in output for same queries submissions [15].

This work presents a clustering-based Web document categorization that faces with positive results, the two fundamental problems of Web clustering: the high dimensionality of the feature space and the knowledge of the entire document. The first problem is tackled with an evolutionary approach. The genetic computation assures stability and efficiency also in presence of a large amount of data. About the second issue we perform a clustering based on the analysis of the context rather than the content of the document. Context-based clustering strongly reduces the size of the Web document to process, without grave fall of performances.

2 A Contextual View of a Web Page

Let us consider a link in a Web page: in general we note the existence of sufficient information spent to describe the referenced page. Thus this information may be used to categorize a document. The process starts with an initial list of URLs, and, for each URL, retrieves the web document, analyzing the structure of the document expressed in terms of its HTML tags. For each meaningful tag, contextual data are extracted. For example, when the <A> tag is found containing an URL, an URL Context Path (URL: C_1: C_2:. . .: C_n) is defined, containing the list of the context strings C_i so far associated to the URL. For example, let us consider the following fragment of an HTML page from Altavista:


networking gateways - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\software\Web Server\Xitami\webpages\Demo\gateways2.h
Go
Collegamenti
Internet

IP Routing
Next: IP Networks Up: Issues of TCP/IP **Networking** Previous: Address Resolution. IP Routing. IP Networks. Subnetworks. **Gateways**. The Routing Table....

Pyx *Networking* Solutions.
Welcome to Pyx **Networking** Solutions. Since 1973, Pyx has been providing quality computer services at great prices to a variety of clients, including...

Satelcom (UK) Limited, home of LAN/WAN Network Integration, Managed Services a
Satelcom (UK) Limited, the leading UK **Networking** & Managed Services Specialist...

Cahners In-Stat Group - Download Residential Gateway White Paper
The convergence of two trends - the demand for high-speed Internet access - and the explosion in interest around home **networking** - has resulted in...

Fig. 1. Example of contexts in a Web page.

The following context paths are created:

1. “http://www.dc.turkuamk.fi/LDP/LDP/nag/node27.html“
   “IP Routing”
   “Next: IP Networks Up: Issues of TCP/IP Networking Previous: Address Resolution. IP Routing. IP Networks. Subnetworks. Gateways. The Routing Table....”
   “Networking Gateways”

2. “http://pyx.net/“:
   “Pyx Networking Solutions. ”
   “Welcome to Pyx Networking Solutions. Since 1973, Pyx has been providing quality computer services at great prices to a variety of clients, including...”
   “Networking Gateways”

3. “http://www.satelcom.co.uk/“:
   “Satelcom (UK) Limited, home of LAN/WAN Network Integration, Managed Services a ”
   “Satelcom (UK) Limited, the leading UK Networking & Managed Services Specialist...”
   “Networking Gateways”

4. “http://www.instat.com/catalog/downloads/resgateway.asp“:
   “Cahners In-Stat Group - Download Residential Gateway White Paper ”
   “The convergence of two trends - the demand for high-speed Internet access - and the explosion in interest around home networking - has resulted in...”
   “Networking Gateways”

Any URL is analyzed through a breadth-first visiting: first the complete page is analyzed, then for each external link a new visiting is triggerred on the corresponding host. Next step regards the clustering process that exploits the Context Paths database and the categories-based catalogue in order to evaluate the membership value of each URL to a set of categories.

3 Architecture

Usually a Web search engine exploits two basic technologies for document retrieval: - *indexing* the Web page is indexed by a number of words or phrases representing an abbreviated version of the page itself; - *directories* the page is represented by a position within a knowledge hierarchy. This section shows how our system enables to classify a Web document with a precision comparable with a directory approach and with a dimensionality and updating speed comparable with an indexing technique. Our system returns a database of the most meaningful categories that characterize a Web area (a set of URLs) under analysis. This task is done thanks to an evolutionary process that updates the previous, existing catalogue.

At instant t_0 we assume the availability of an initial catalogue, used as a kind of training set. The evolved catalogue, containing new category entries, is then used to classify the Web documents. The system is based on a client-server architecture in order to distribute the computational agents charged to load the document from the Web and to classify the document itself.

The evolution layer consists of different modules: (1) on the client-side the *SpiderAgents* have been implemented in order to acquire the context paths of the Web documents, (2) on the server-side the software agents *Genetic Engine*

have been realized in order to collect the context paths and to transform them into genotypes. This enables to produce, through the genetic-based process, the catalogue, and (3) the agents *Clusterizer* has been designed to classify the Web documents.
Here follows a short discussion about the basic technologies employed for the automatic categorization.

Spidering: the goal of the spidering process is to perform a parsing of the document in order to extract the information concerning the context paths;

Classification: we use a model of *context fuzzy clustering*, based on syntax analysis (part of speech) and semantic analysis (WordNet [18]) of the information derived from the context paths;

Evolution of the category catalogue: the context fuzzy clustering is embedded into a genetic framework able to produce automatically an updating procedure on the catalogue.

The system is written in Java 2 [16], the distributed computation is managed using **Remote Method Invocation** (**RMI**) technology supported by the SUN platform **JDK**.

4 Clustering Methodology

Let **T** be the set of the noun phrases. $\forall x \in T$ we define $\widetilde{x}$ as the *fuzzy set* "noun phrases *similar to* x", formally:

$$\widetilde{\mathbf{x}} = \{(t, \mu_x(t)) \mid \forall t \in T\}$$

with $\mu_x : T \to [0,1]$ as membership function.

The function is defined in order to give higher values for the noun phrase that generalizes the original term of the category. The function takes into account the synonyms for each simple term contained into the noun phrase of the category, rejecting the terms that are not synonyms or related terms. Any synonym of the simple term has a weight: the weights are higher for hypernym synonyms (generalization terms) and lower for hyponym synonyms (specialization terms), hence the clustering method brings up generalization with respect to each document matched. The membership value of a noun phrase, derived from a combination of simple terms, is given as an average of the synonyms weights.

Given **P(T)** as the power set of T, let us define the following similarity measure:

Let $x = (t_1, \ldots, t_n) \in P(T)$ and $t_i \in T\ \forall i = 1..n$
$y = (h_1, \ldots, h_p) \in P(T)$ and $h_j \in T\ \forall j = 1..p$

$$\mathbf{S_K}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{p} \sum_{i=1}^{n} (\mu_{t_i}(h_j))^K \qquad \text{(shortly } x \oplus_k y\text{)} \tag{1}$$

where **K** is the *similarity factor* of the measure.

Given a couple $(x, y) \in P(T)^2$ we define **G**: P(T)xP(T)→ [0, 1] as the *coverage* of y on x:

$$\mathbf{G}(\mathbf{x}, \mathbf{y}) = \frac{|\{h_j|\ h_j \in\ y\ and\ \exists\ t_i \in\ x\ \ \ni'\ \mu_{t_i}(h_j) > 0\}|}{|x|} \qquad \text{(shortly } x \sqcap y\text{)} \tag{2}$$

Each category (or sub-category), defined by its noun phrases, is viewed as a cluster $\mathbf{C_j} \in \mathbf{P(T)}$. Objects of the cluster are URLs extracted from the Web documents: each URL has an associated **Context Path** as *feature vector*, represented by $\mathbf{CP_i} \in P(T)$ (for the i^{th} context path).

In order to evaluate the membership grade $\mu_{\mathbf{ij}}$ of the $\mathbf{CP_i}$ on cluster $\mathbf{C_j}$, a *familiarity grade* $\mathbf{A_{ij}}$ is defined; this parameter is the weight returned by the matching between context path and category, computed as the similarity measure on $\mathbf{P(T)}$ between $\mathbf{C_j}$ and $\mathbf{CP_i}$.

Up now the clusters are statically defined (their noun phrases are fixed). The dynamical behavior is provided by the genetic exploration (as defined in the next paragraph) and by a *specialization grade* **s** for each cluster, that allows us to vary the cluster dimension. The specialization grade exploits the *similarity factor* ***K*** that enables to modify the incidence of each similarity grade for the single terms. The next formula defines the familiarity grade using the specialization grade s_j for cluster C_j.

Familiarity Grade:

$$A_{ij} = \frac{C_j\ \oplus_{s_j}\ CP_i}{noun\ phrases\ matched\ by\ CP_i\ on\ C_j} \tag{3}$$

$A_{ij} \in [0,\ 1]$

Membership Grade:

$$\mu_{ij} = A_{ij} \cdot (C_j\ \sqcap\ CP_i)\ \ \mu_{ij} \in [0,\ 1] \tag{4}$$

Our clustering method exploits the concept of the *overlapping* flexibility; it allows objects to belong to all clusters.

Overlapping Property:

$$\sum_{j=1}^{|C|} \mu_{ij} \geq 0 \tag{5}$$

Finally, the clustering method maximizes the following *Index of Quality* $\boldsymbol{J(C)}$, for which an *Influence Grade* **m** is introduced in order to reduce the impact of lower μ_{ij} values. At the increasing of **m** more relevant will be the weight of the clusters characterized by a higher specialization (membership grade).

Index of Quality:

$$J(C) = \sum_{j=1}^{C} (J_j) \tag{6}$$

$$J_j = \begin{cases} (\sum_{i=1}^{N} \mu_{ij})^m & \text{no subcategory in } C_j \\ ((\sum_{i=1}^{N} \mu_{ij} + 1) \cdot \sum_{c}^{subcategs\ C_j} J_c)^m & \text{otherwise} \end{cases} \tag{7}$$

with $m \in [1,\infty)$ and J_j as Index of Quality for the j^{th} category.

Index of Quality is skilled to specialize the categories, in order to contrast the generalization spur arising from the computation of matching weights.

5 Genetic Framework

1. **Representation of genomes** – the genome is defined through a tree-based structure, namely *Category Forest*, introduced as a hierarchical model of the thematic categories. Each category is represented by a *Category Tree.* A Category Tree is identified by a *Root Category* representing a main topic. Starting from a Root Category we find the subcategory nodes (specialization of a topic) which, in their turn, may be parents of more specific topics as shown in the Figure 2.

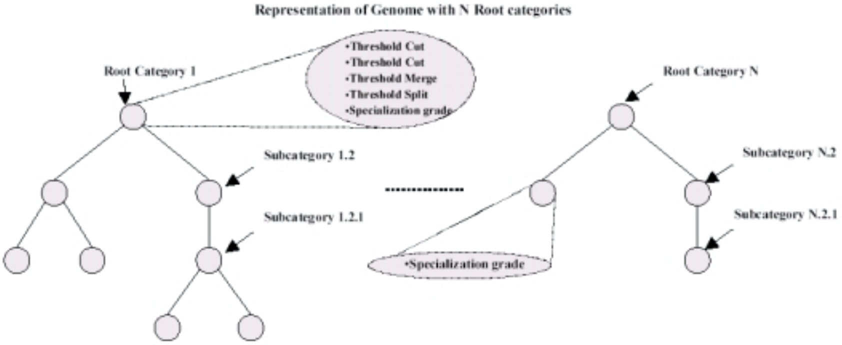


Fig. 2. Representation of Genomes.

Each root node is supported by three threshold values useful to handle the specialization grade of the thematic category.
The subcategories can be defined **fixed** in the parent category, by means of a marker; this is useful to do not move the subcategory into other parent categories as effect of the mutation operator.

2. **Definition of the fitness function** - Two different evaluations are introduced. The first, named *Clustering Fitness*, is computed by the clustering methodology in terms of Index of Quality. The second factor is the *Quality of*

Distribution (QoD), measuring the quality of distribution of the document into thematic categories. This value is computed by averaging the membership grades of the document, for each category or subcategory.

- **Clustering Fitness (Index of Quality)**, see the formulas (6) and (7)
- **Quality of Distribution (QoD):**

$$QoD = \frac{\sum(QoD_{category})}{\#root\ categories}$$

$$QoD_{category} = \frac{\widehat{\mu}+\sum(QoD_{category}\ of\ the\ subcategories)}{\#subcategories+1}$$

  where $\widehat{\mu}$ is the average of membership values of the document into the category (root category or subcategory).
- **Fitness function of the individual:**

$$Fitness = QoD * ClusteringFitness$$

3. **Definition of the Crossover operator** – The crossover point is chosen randomly taking into account the root categories that can not be broken by crossover.
4. **Definition of mutation operators** – The following mutation operators are defined:
   - **Mutation Cutting** – Choose randomly both a root category and a subcategory into it: the subcategory is removed together with its subtree.
   - **Mutation Merging** – Choose randomly a root category and extract randomly two "sister" subcategories (nodes with the same parent category). The operator merges the root nodes of the two selected subcategories.
   - **Mutation Specialization Grade**– Choose randomly a root category and modify its specialization grade.
   - **Mutation Exchange Parent (Swap)** – Choose randomly a root category and extract randomly two subcategories with different parent categories. Hence, the operator swaps the parent categories.
   - **Mutation Change Parent** – Choose randomly both a root category and a subcategory. Hence, the operator moves the subtree in another parent category randomly.

6 Testing

In order to verify the efficiency of our clustering methodology we take as target the Open Directory Project(ODP) [9] a well known (public domain) project of human categorization of Web documents. We use the synonyms and related terms, computed in advance for each category of the catalogue, using WordNet [18].

Our experiment has been conducted on the following subset of the categories catalogue of ODP :

Science
Business
Private
Reference
Society
Health
Test
World
Shopping
Recreation
Arts
Home
Computers
Games
Bookmarks
Sports
Regional
News

The URLs, with their short description, are collected in an HTML document in order to extract the corresponding Context Paths.

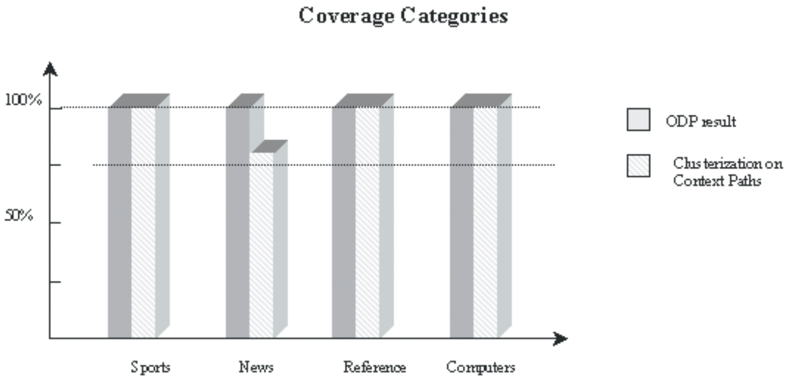


Fig. 3. Coverage Categories.

Figure 3 reports the behavior of our approach compared with ODP. We obtained automatically the same ”human” categorization for the categories Sports, Reference and Computer.

As shown in the figure, the ”News” category is not totally covered. This happens because into this category there are URLs not completely described. Below we give an example of context paths of some URLs (contained into ODP database) that our clustering is not able to associate to the right ”News” category.

“http://www.bcity.com/bollettino:“
  “International Bulletin”
    “International politics. Italian, French and (some) English.”

“http://www.pressdigest.org/“:
  “Pressdigest”
    “International and Multilingual press digest.”

The reason of this drawback is due to the WordNet database: the term ”news” is not related to ”bulletin” and ”digest” as synonyms.

In order to highlight the role of fuzziness, Figure 4 shows the membership value of the URL http://attention.hypermart.net associated to the category ”News”.

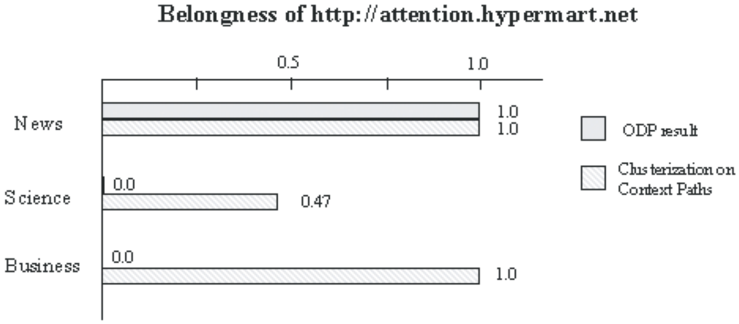


Fig. 4. Belongness example.

As noted in Figure 4, the URL is not exclusively associated to the category (as opposite of ODP): this means that in our case, if the user searches URLs about technology into the category ”Science” , the search engine shall be able to return a reference to the URL attention.hypermart.net, even though with a membership value lower than the News category.

7 Related Works

The role of cluster, as useful strategy to improve Web search engine behaviors, has reported an increasing interest in these recent years. A well explored issue is to cluster the results of a Web search to better formulate the query. In [4] the query refinement, obtained also thanks to the user’s feedback, guarantees a customization of a search space that better fits the user’s need. In [2] it is proved how a graph partitioning based clustering technique, without the constraint to specify pre-specified ad-hoc distance functions, can effectively discover Web document similarities and associations. A linear time algorithm which creates clusters on the analysis of phrases shared between Web documents is discussed in [17]. A machine learning approach has been used in [12] and [6] for efficient topic-directed spidering and relevant topic extraction. A fuzzy matching for information retrieval searching is discussed in [5].
About the use of contextual information, the ARC system [3] automatically compiles a list of authoritative Web resources on a topic. [1] is the first concrete effort of a context-based categorization even though the methodology does not support fuzzy partitioning and the search of the better partitioning could suffer of the usual drawbacks concerning traditional clustering algorithms.

Conclusions

In this paper, we present a methodology able to cluster web document into thematic categories. The clustering algorithm is based on a fuzzy clustering method that searches the best categories catalogue for web document categorization. The categorization is performed by context, this means that the clustering is guided by the context surrounding a link in an HTML document in order to extract useful information for categorizing the document it refer to. This approach enables to be media independent, hence to perform the same strategy also for images, audio and video. As key issue of our clustering methodology we use an evolutionary approach inheriting the benefits of a genetic-level explorations. The positive benchmarks reported by comparing our results with a public-domain, significant category-based catalogue stimulates further development of our research.

References

1. Attardi, G., Di Marco S., and Salvi, D. (1998). Categorisation by Context. *Journal of Universal Compouter Science*, 4:719-736.
2. Boley, D., Gini., M., Gross, R., Hang, E-H., Hasting, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. (1999). Partioning-based clustering for Web document categorization *Decision Support System*, 27 (1999) 329-341.
3. Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Rahavan, P., and Rajagopalan, S.(1998). Automatic resource list compilation by analyzing hyperlink structure and associated text. *Seventh International World Wide Web Conference*, 1998.
4. Chang, C-H., and Hsu, C-C. (1997). Customizable Multi-Engine Search tool with Clustering. *Sixth International World Wide Web Conference*, April 7-11, 1997 Santa Clara, California, USA.
5. Cohen, W. (1998). A web-based information system that reasons with structured collections of text. *Agents'98*, 1998.
6. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. (1998). Learning to extract symbolic knowledge from the World Wide Web. *AAAI-98*, 1998.
7. Hayes, J., and Weinstein, S. P. (1990). CONSTRUE-TIS: A system for content-based indexing of a database of news stories. *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1-5.
8. Iwayama, M. (1995). Cluster-based text categorization : a comparison of category search strategies. *SIGIR-95*, pp. 273-280.
9. Open Directory Project. URL: http://dmoz.org/about.html
10. Lawrence, S. and Giles, C. L. (1999). *Nature*, 400:107-109. *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99).*
11. Mase, H., Tsuji, H., Kinukawa, H., Hosoya, Y., Koutani, K., and Kiyota, K. (1996). Experimental simulation for automatic patent categorization. *Advances in Production Management Systems*, 377-382.
12. McCallum, A., Nigam, K., Rennie, J., and Seymore, K. (1999). A Machine Learning Approach to Building Domain-Specific Search Engine. *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99).*
13. Sahami, M., Yusufali, S., and Baldoando, M. Q., W. (1998) SONIA: A service for organizing networked information autonomously. *Third ACM Conference on Digital Libraries.*

14. Selberg, E. (1999) *Towards Comprehensive Web Search.* PhD thesis, University of Washington.
15. Selberg,E and Etzioni, O. (2000). On the Instability of Web Search Engine. *RIAO 2000.*
16. JDK Java 2 Sun. http://java.sun.com
17. Zamir, O., and Etzioni, O. (1988). Web Document Clustering: A Feasibility Demonstration. *SIGIR'98*, Melbourne, Australia, ACM Press.
18. A Lexical Database for English. URL: http://www.cogsci.princeton.edu/ wn/