Classifying Web Data in Directory Structures

Sofia Stamou[1], Alexandros Ntoulas[2], Vlassis Krikos[1], Pavlos Kokosis[1],
and Dimitris Christodoulakis[1]

[1] Computer Engineering and Informatics Department, Patras University,
26500 Patras, Greece
{stamou, krikos, kokosis}@ceid.upatras.gr
dxri@upatras.gr
[2] Computer Science Department, University of California,
Los Angeles (UCLA), USA
ntoulas@cs.ucla.edu

Abstract. Web Directories have emerged as an alternative to the Search Engines for locating information on the Web. Typically, Web Directories rely on humans putting in significant time and effort into finding important pages on the Web and categorizing them in the Directory. In this paper, we experimentally study the automatic population of a Web Directory via the use of a subject hierarchy. For our study, we have constructed a subject hierarchy for the top level topics offered in Dmoz, by leveraging ontological content from available lexical resources. We first describe how we built our subject hierarchy. Then, we analytically present how the hierarchy can help in the construction of a Directory. We also introduce a ranking formula for sorting the pages listed in every Directory topic, based on the pages' quality, and we experimentally study the efficiency of our approach against other popular methods for creating Directories.

1 Introduction

Web Directories have emerged as an alternative to the well-established Web Search Engines, for locating information on the Web. Typically, a Web Directory, e.g. the Dmoz Directory [2], organizes Web pages in a subject hierarchy and allows users to locate interesting information by navigating through the hierarchy. Despite the simplicity of navigating in the contents of Web Directories, their editing and maintenance are tedious and time-consuming, since the task of assigning Web pages to topic Directories relies exclusively on the indispensable effort of human editors. However, the sheer quantity of information that is available on the Web restrains the exhaustive investigation of each and every Web page before these are assigned to topical categories. To make things worse, the staggering rates of Web's evolution [22] get humans overwhelmed by the amount of data that they need to painstakingly examine and categorize within the Directories' contents. Clearly, if we could help Web editors automate their task we would save a lot of time for a number of people.

One way to alleviate the problem of categorizing Web pages inside a Directory's topics is to employ machine learning techniques in order to build a classifier, which will then assign every Web page to a topic. However, this approach, requires a considerable number of training examples to build accurate classifiers, and might prove

X. Zhou et al. (Eds.): APWeb 2006, LNCS 3841, pp. 238 – 249, 2006.

inefficient for Web scale classification. This is due to the Web's dynamic nature, which imposes the need for re-training the classifier (possibly on a new dataset) every time a change is made.

In this paper, we present an alternative approach for the effective population of Web Directories, which does not require training and, therefore, it can cope easily with changes on the Web. The only input that our method requires is a subject hierarchy that one would like to use and a collection of Web pages that one would like to assign to the hierarchy's subjects. Besides the automatic population of Web Directories, our approach offers an efficient way of ordering the Web pages inside the Directory's topics, by ranking the pages based on how "descriptive" they are of the category they are assigned to. At a high level our method proceeds as follows: First, we leverage ontological content from freely available resources created by the Natural Language Processing community, in order to build a subject hierarchy. Then, given a collection of Web pages, we pre-process them in order to extract the words that "best" communicate every page's theme, the so-called thematic words. We use the pages' thematic words and the hierarchy to compute one or more subjects to assign to every page. Moreover, we employ a ranking algorithm, which measures the pages' closeness to the subjects, as well as the semantic correlations among the pages in the same subject, and sorts the pages listed in each Directory topic, so that pages of good quality appear earlier in the results.

In Section 5, we experimentally evaluate the performance of our approach in categorizing a sample of nearly 320,000 Web pages and we compare it to the performance of other classification schemes. Obtained results show that the categorization accuracy of our automatic classification method is comparable to the accuracy of machine learning classification techniques. However, in our classification method, no training set is required.

We start our discussion by presenting the subject hierarchy that we developed for the top level topics used in a popular Web Directory, i.e. Dmoz. Then, in Section 3, we describe how we identify thematic words inside every Web page and we show how we employ thematic words to assign Web pages to the hierarchy's subjects. In Section 4, we introduce a ranking formula, which sorts the pages listed in every Directory topic by prioritizing pages of higher classification accuracy. Our experimental results are presented in Section 5 and we conclude our work in Sections 6 and 7.

2 A Subject Hierarchy for the Web

Web Directories offer a browsable topic hierarchy that is used for organizing Web pages into topics. Currently, topic hierarchies are constructed and maintained by human editors, who manually locate interesting Web pages. Based on the pages' content, the editors find the best fit for the page among the hierarchy's topics. Apparently, the manual construction of Web Directories is tedious and may suffer from inconsistencies. To overcome the difficulties associated with editing Web Directories, we built a topic hierarchy, which we use for automatically categorizing Web pages.

Our hierarchy essentially integrates domain information from the Suggested Upper Merged Ontology (SUMO) [3] and the MultiWordNet Domains (MWND) [1], into WordNet 2.0 [4]. Since a fraction of WordNet's hierarchies is already annotated with domain information, our task was essentially to anchor a domain label to the remaining hierarchies. To that end, we firstly anchored to those WordNet hierarchies that are uniquely annotated in either SUMO or MWND their corresponding domain labels. In selecting a domain label, for the hierarchies that are assigned a different domain between SUMO and MWND, we merged those hierarchies together and we picked the domains of the merged hierarchies' parent nodes. Merging was generally determined by the semantic similarity that the concepts of the distinct hierarchies exhibit, where semantic similarity is defined as a correlation of: (i) the length of the path that connects two concepts in the shared hierarchy and (ii) the number of common concepts that subsume two concepts in the hierarchy [25]. Lastly, we attached to each of the hierarchy's lower level concepts those WordNet hierarchies that encounter a specialization (is-a) relation to it. A detailed description of the process we followed for building our hierarchy can be found in [11].

To demonstrate the usefulness of our hierarchy in a real-world setting, we augmented the hierarchy with topics that are currently used by Web cataloguers for classifying Web data. For that purpose, we explored the first level topics in the Dmoz Directory and, using WordNet, we selected the Dmoz topics that are super-ordinates of the merged hierarchies' root concepts. The selected Dmoz topics (shown in Table 1) were incorporated, through the is-a relation, in our hierarchy and formed the hierarchy's first level topics.

Table 1. The hierarchy's first-level topics

Topics in our Hierarchy	
Arts	Health
Sports	News
Games	Society
Science	Computers
Reference	Home
Shopping	Recreation
Business	

At the end of this merging process, we came down to a hierarchy of 489 concepts that are organized into 13 topics. The resulting hierarchy is a directed acyclic graph where each node represents a concept, denoted by a unique label, and linked to other concepts via a specialization (is-a) link. The maximum depth of the hierarchy's graph is 4 and the maximum number of children concepts (i.e. branching factor) from a node is 26. An important note here is that our hierarchy can be tailored to accommodate any first level topics that one would like to use, as long as these are represented in WordNet. In addition, the hierarchy could be used in a multilingual setting, through the use of aligned WordNets [29].

3 Finding Web Pages' Thematic Words

The main intuition in our approach for categorizing Web pages is that topic relevance estimation of a page relies on the page's lexical coherence, i.e. having a substantial portion of words associated with the same topic. To capture this property, we adopt the lexical chaining approach and, for every page, we generate a sequence of semantically related terms, known as lexical chain. The computational model we used for generating lexical chains is presented in the work of [6] and it generates lexical chains in a three-step process: (i) select a set of candidate terms[1] from the page, (ii) for each candidate term, find an appropriate chain relying on a relatedness criterion among members of the chains, and (iii) if it is found, insert the term in the chain. The relatedness factor in the second step is determined by the type of WordNet links that connect the candidate term to the terms stored in existing chains. We then disambiguate the words inside every generated lexical chain, using the scoring function *f* introduced in [27], which indicates the possibility that a word relation is a correct one.

Given two words, w_1 and w_2, their scoring function *f* via a relation *r*, depends on the words' association score, their depth in WordNet and their respective relation weight. The association score (*Assoc*) of the word pair (w_1, w_2) is determined by the words' corpus co-occurrence frequency and it is given by:

$$Assoc(w_1, w_2) = \frac{\log\,(p\,(w_1, w_2) + 1)}{N_s\,(w_1)\bullet N_s\,(w_2)}\,. \tag{1}$$

where, $p(w_1,w_2)$ is the corpus co-occurrence probability of the word pair (w_1,w_2) and $N_s(w)$ is a normalization factor, which indicates the number of WordNet senses that word w has. Given a pair (w_1,w_2), their *DepthScore* expresses the words' position in WordNet hierarchy and is defined as:

$$DepthScore(w_1, w_2) = Depth(w_1)^2 \bullet Depth(w_2)^2. \tag{2}$$

where, *Depth*(w) is the depth of word w in WordNet. Semantic relation weights (*RelationWeight*) have been experimentally fixed to 1 for reiteration, 0.2 for synonymy and hyper/hyponymy 0.3 for antonymy, 0.4 for mero/holonymy and 0.005 for siblings. The scoring function *f* of w_1 and w_2 is defined as:

$$f_s(w_1, w_2, r) = Assoc\,(w_1, w_2)\bullet DepthScore(w_1, w_2)\bullet \mathrm{Re}\,lationWeight(r). \tag{3}$$

The score of the lexical chain C_i that comprises w_1 and w_2, is calculated as the sum of the score of each relation r_j in C_i. Formally:

$$Score(C_i) = \sum_{r_j\ in\,C_j} f_s\ (w_{j1}, w_{j2}, r_{\ j})\,. \tag{4}$$

To compute a single lexical chain for every downloaded Web page, we segment the latter into shingles [8], and for every shingle, we generate scored lexical chains, as described before. If a shingle produces multiple chains, the lexical chain of the highest score is considered as the most representative chain for the shingle. In this way, we eliminate chain ambiguities. We then compare the overlap between the elements of all shingles' lexical chains consecutively. Elements that are shared across chains are deleted so that lexical chains display no redundancy. The remaining elements are merged together into a single chain, representing the contents of the entire page, and a new *Score*(C_i) for the resulting chain C_i is computed.

[1] Candidate terms are nouns, verbs, adjectives or adverbs.

3.1 Categorizing Web Pages

In order to assign a topic to a Web page, our method operates on the page's thematic words. Specifically, we map every thematic word of a page to the hierarchy's topics and we follow the hierarchy's hypernymic links of every matching topic upwards until we reach a root node. For short documents with very narrow subjects this process might yield only one matching topic. However, due to both the great variety of the Web data and the richness of the hierarchy, it is often the case that a page contains thematic words corresponding to multiple root topics.

To accommodate multiple topic assignment, a *Relatedness Score* (*RScore*) is computed for every Web page to each of the hierarchy's matching topics. This *RScore* indicates the expressiveness of each of the hierarchy's topics in describing the pages' content. Formally, the *RScore* of a page represented by the lexical chain C_i to the hierarchy's topic D_k is defined as the product of the chain's *Score*(C_i) and the fraction of the chain's elements that belong to topic D_k. We define the *Relatedness Score* of the page to each of the hierarchy's matching topics as:

$$\text{RScore (i, k)} = \frac{\text{Score}(C_i) \bullet \#\text{of } C_i \text{ elements of } D_k \text{ matched}}{\left|\#\text{of } C_i \text{ elements}\right|}. \tag{5}$$

The denominator is used to remove any effect the length of a lexical chain might have on *RScore* and ensures that the final score is normalized so that all values are between 0 and 1, with 0 corresponding to no relatedness at all and 1 indicating the category that is highly expressive of the page's topic. Finally, a Web page is assigned to the topical category D_k for which it has the highest relatedness score of all its *RScores* above a threshold T, with T been experimentally fixed to T= 0.5. The page's indexing score is:

$$\textit{IScore}\text{ (i, k)} = \max \textit{RScore}\text{ (i, k)}. \tag{6}$$

Pages with chain elements matching several topics in the hierarchy, and with relatedness scores to any of the matching topics below T, are categorized in all their matching topics. By allowing pages to be categorized in multiple topics, we ensure there is no information loss during the Directories' population and that pages with short content (i.e. short lexical chains) are not unquestionably discarded as less informative.

4 Organizing Web Pages in Directories

Admittedly, the relatedness score of a page to a Directory topic does not suffice as a measurement for ordering the pages that are listed in the same Directory topic. This is because *RScore* is not a good indicator of the amount of content that these pages share. Herein, we report on the computation of semantic similarities among the pages that are listed in the same Directory topic. Semantic similarity is indicative of the pages' correlation and helps us determine the ordering of the pages that are deemed related to the same topic.

To estimate the semantic similarity between a set of pages, we compare the elements in a page's lexical chain to the elements in the lexical chains of the other pages in a Directory topic. Our intuition is that the more elements the chains of two pages have in common, the more correlated the pages are to each other. To compute similarities between pages, P_i and P_j that are assigned to the same topic, we first need to identify the common elements between their lexical chains, represented as PC_i and PC_j respectively Then, we use the hierarchy to augment the elements of the chains PC_i and PC_j with their synonyms. Chain augmentation ensures that pages of comparable content are not regarded unrelated if their lexical chains contain distinct but semantically equivalent elements (i.e. synonyms). The augmented elements of PC_i and PC_j respectively, are defined as:

$$\begin{gathered} AugElements(PC_i) = C_i \bigcup Synonyms(C_i) \text{ and} \\ AugElements(PC_j) = C_j \bigcup Synonyms(C_j) \end{gathered} \tag{7}$$

where, *Synonyms* (C_i) denotes the set of the hierarchy's concepts that are synonyms to any of the elements in C_i and *Synonyms* (C_j) denotes the set of the hierarchy's concepts that are synonyms to any of the elements in C_j. The common elements between the augmented lexical chains PC_i and PC_j are determined as:

$$ComElements(PC_i, PC_j) = AugElements(PC_i) \bigcap AugElements(PC_j) \, . \tag{8}$$

We formally define the problem of computing pages' semantic similarities as follows: if the lexical chains of pages p_i and p_j share elements in common, we produce the correlation look up table with tuples of the form <*AugElements* (*PC*$_i$), *AugElements (PC*$_j$), *ComElements*>. The similarity measurement between the lexical chains PC_i, PC_j of the pages P_i and P_j is given by:

$$\sigma_s(PC_i, PC_j) = \frac{2 \bullet \left| ComElements(PC_i, PC_j) \right|}{\left| AugElements(PC_i) \right| + \left| AugElements(PC_j) \right|} \, . \tag{9}$$

where, the degree of semantic similarity is normalized so that all values are between zero and one, with 0 indicating that the two pages are totally different and 1 indicating that the two pages talk about the same thing.

4.1 Ranking Web Pages in Directory Topics

We sort the pages assigned to a Directory topic, in terms of a DirectoryRank (*DR*) metric, which estimates the "importance" of pages in a Directory. DirectoryRank is inspired by, and thus resembles, the PageRank measure [23] in the sense that the importance of a page is high if it is somehow connected to other important pages, and that important pages are valued more highly than less important ones. While PageRank realizes the connection between pages in terms of their in/out-going links to other pages, DirectoryRank defines the connection between pages in terms of their semantic coherence to other pages in the Directory, this is; it estimates the importance of pages from their degree of semantic similarity to other important pages.

Intuitively, an important page in a Directory topic, is a page that has a high relatedness score to the Directory's topic and that is semantically close (similar) to many other pages in that topic. *DR* defines the quality of a page to be the sum of its topic relatedness score and its overall similarity to the fraction of pages with which it correlates in the given topic. This way, if a page is highly related to topic *D* and also correlates highly with many informative pages in *D*, its *DR* score will be high. Formally, consider that page p_i is indexed in Directory topic T_k with some *RScore* (p_i, T_k) and let p_1, p_2, …, p_n be pages in T_k with which p_i semantically correlates with scores of σ_s (PC_1, PC_i), σ_s (PC_2, PC_i),…, σ_s (PC_n, PC_i), respectively. Then, the DirectoryRank (*DR*) of p_i is given by:

$$DR\,(p_i,T_k) \;=\; RScore\,(p_i,T_k) + \;[\sigma_s\,(PC_1,PC_i) + \sigma_s\,(PC_2,PC_i) + \ldots\ldots + \sigma_s\,(PC_n,PC_i)]\,/\,n\,. \qquad (11)$$

where *n* corresponds to the total number of pages in topic T_k with which p_i semantically correlates. High *DR* values imply that: (i) there are some “good quality” sources among the data stored in a Directory, and that (ii) more users are likely to visit them while browsing the Directory’s contents. Summarizing, the DirectoryRank metric determines the ranking order of the pages associated with a Directory and serves towards giving higher rankings to the more “important” pages of the Directory.

5 Evaluation of Automatic Categorization

To study the effectiveness of our method in automatically assigning Web pages into a subject hierarchy, we run an experiment where we compared the efficiency of our method in categorizing Web pages in the Dmoz topics, to the efficiency of a Naïve Bayesian classifier in categorizing the same set of pages in the same topics.

5.1 Experimental Setup

In selecting our experimental data, we wanted to pick a useful yet representative sample of the Dmoz’s content. By useful, we mean that our sample should comprise Web pages with textual content and not only links, frames or audiovisual data. By representative, we mean that our sample should span those Dmoz’s categories, whose topics are among the top level topics in our subject hierarchy.

To obtain such a sample, we downloaded a set of 318,296 Web pages listed in the 13 Dmoz topics that are represented in our hierarchy. We parsed the downloaded pages and generated their shingles, after removing HTML markup. Pages were then tokenized, part-of-speech tagged, lemmatized and submitted to our classification system, which following the process described above, computed and weighted a single lexical chain for every page. To compute lexical chains, our system relied on a resources index, which comprised (i) the 12.6M WordNet 2.0 data for determining the semantic relations that exist between the pages’ thematic words, (ii) a 0.5GB compressed TREC corpus from which we extracted a total of 340MB binary files for obtaining statistics about word co-occurrence frequencies, and (iii) the 11MB top level concepts in our hierarchy. Table 2 shows some statistics of our experimental data. Our system generated and scored simple and augmented lexical chains for every page and based on a combined analysis of this information it indicates the most appropriate topic in the hierarchy to categorize each of the pages.

To measure our system’s effectiveness in categorizing Web pages, we experimentally studied its performance against the performance of a Naïve Bayes classifier, which has proved to be efficient for Web scale classification [14]. In particular, we

Table 2. Statistics on the experimental data

Category	# of documents	Average # of shingles
Arts	28,342	30
Sports	20,662	13
Games	11,062	17
Home	6,262	11
Shopping	52,342	12
Business	60,982	16
Health	23,222	25
News	9,462	37
Society	28,662	45
Computers	35,382	25
Reference	13,712	33
Recreation	8,182	19
Science	20,022	32
Total	318,296	

trained a Bayesian classifier by performing a 70/30 split to our experimental data and we used the 70% of the downloaded pages in each Dmoz topic as a learning corpus. We then tested the performance of the Bayesian classifier in categorizing the remaining 30% of the pages in the most suitable Dmoz category. For evaluating the classification accuracy of both the Bayesian and our classifier, we used the Dmoz categorizations as a comparison testbed, i.e. we compared the classification delivered by each of the two classifiers to the classification done by the Dmoz cataloguers for the same set of pages. Although, our experimental pages are listed in all sub-categories of the Dmoz's top level topics, for the experiment presented here, we focus on classifying the Web pages only for the top-level topics.

5.2 Discussion of the Experimental Results

The overall accuracy results are given in Table 3, whereas Table 4 compares the accuracy rates for each category between the two classifiers. Since our classifier allows pages with low *RScores* to be categorized in multiple topics, in our comparison we explored only the topics of the highest *RScores*. Note also that we run the Bayesian classifier five times on our data, every time on a random 70/30 split and we report on the best accuracy rates among all runs for each category.

The overall accuracy rates show that our method has improved classification accuracy compared to Bayesian classification. The most accurate categories in our classification method are ***Arts*** and ***Society***, which give 90.70% and 88.54% classification accuracy respectively. The underlying reason for the improved accuracy of our classifier

Table 3. Overall accuracy results from both classifiers

Classifier	Accuracy	Standard Error Rate
Bayesian	65.95%	0.06%
Ours	69.79%	0.05%

Table 4. Comparison of average accuracy rates between categories for the two classifiers

Category	Bayesian classifier	Our classifier
Arts	67.18%	90.70%
Sports	69.71%	75.15%
Games	60.95%	64.51%
Home	36.56%	40.16%
Shopping	78.09%	71.32%
Business	82.30%	70.74%
Health	64.18%	72.85%
News	8.90%	55.75%
Society	61.14%	88.54%
Computers	63.91%	74.04%
Reference	20.70%	69.23%
Recreation	54.83%	62.38%
Science	49.31%	71.90%

in those topics is the fact that our hierarchy is rich in semantic information for those topics. This argument is also attested by the fact that for the topics ***Home*** and ***News***, for which our hierarchy contains a small number of lexical nodes, the classification accuracy of our method is relatively low, i.e., 40.16% and 55.75% respectively. Nevertheless, even in those topics our classifier outperforms the Bayesian classifier, which gives for the above topics a classification accuracy of 36.565% and 8.90%. The most straightforward justification for the Bayesian's classifier low accuracy in the topics ***Home*** and ***News*** is the limited number of pages that our collection contains about those two topics. This is also in line with the observation that the Bayesian classifier outperforms our classifier when (i) dealing with a large number of documents, and/ or (ii) dealing with documents comprising specialized terminology. The above can be attested in the improved classification accuracy of the Bayesian classifier for the categories ***Business*** and ***Shopping***, which both have many documents and whose documents contain specialized terms (e.g. product names) that are underrepresented in our hierarchy.

A general conclusion we can draw from our experiment is that, given a rich topic hierarchy, our method is quite promising in automatically classifying pages and incurs little overhead for Web-scale classification. While there is much room for improvement and further testing is needed before judging the full potential of our method, nevertheless, based on our findings, we argue that the current implementation of our system could serve as a Web cataloguers' assistant by delivering preliminary categorizations for Web pages. These categorizations could be then further examined by human editors and reordered when necessary. Finally, in our approach, we explore the pages' classification probability (i.e. *RScore*) so that, upon ranking, pages with higher *RScores* are prioritized over less related pages. This, in conjunction with the pages' semantic similarities, forms the basis of our ranking formula (DirectoryRank). An early study about the potential of DirectoryRank can be found in [28].

6 Related Work

The automated categorization of Web documents into pre-defined topics has been investigated in the past. Previous work mainly focuses on using machine learning techniques to build text classifiers. Several methods have been proposed in the literature for the construction of document classifiers, such as decision trees [5], Support Vector Machines [13], Bayesian classifiers [24], hierarchical text classifiers [19], [11], [9], [20], [26], [12], [21], [7], [17]. The main commonality in previous methods is that their classification accuracy depends on a training phase, during which statistical techniques are used to learn a model based on a labeled set of training exampled. This model is then applied for classifying unlabeled data. While these approaches provide good results, they are practically inconvenient for Web data categorization, mainly because it is computationally expensive to continuously gather training examples for the ever-changing Web. The distinctive feature in our approach from other text classification techniques is that our method does not require a training phase, and therefore it is convenient for Web scale classification.

An alternative approach in categorizing Web data implies the use of the Web pages' hyperlinks and/or anchor text in conjunction with text-based classification methods [10], [15], [16]. The main intuition in exploring hypertext for categorizing Web pages relies on the assumption that both the links and the anchor text of Web pages communicate information about the pages' content. But again, classification relies on a training phase, in which labeled examples of anchor text from links pointing to the target documents are employed for building a learning model. This model is subsequently applied to the anchor text of unlabeled pages and classifies them accordingly. Finally, the objective in our work (i.e. populating Web Directories) could be addressed from the agglomerative clustering perspective; a technique that treats the generated clusters as a topical hierarchy for clustering documents [18]. The agglomerative clustering methods build the subject hierarchy at the same time as they generate the clusters of the documents. Therefore, the subject hierarchy might be different between successive runs of such an algorithm. In our work, we preferred to build a hierarchy by using existing ontological content, rather than to rely on newly generated clusters, for which we would not have perceptible evidence to support their usefulness for Web data categorization. However, it would be interesting for the future to take a sample of categorized pages and explore it using an agglomerative clustering module.

7 Concluding Remarks

We have presented a method, which uses a subject hierarchy to automatically categorize Web pages in Directory structures. Our approach extends beyond data classification and challenges issues pertaining to the Web pages' organization within Directories and the quality of the categorizations delivered. We have experimentally studied the effectiveness of our approach in categorizing a fraction of Web pages into topical categories, by comparing its classification accuracy to the accuracy of a Bayesian classifier. Our findings indicate that our approach has a promising potential in facilitating current tendencies in editing and maintaining Web Directories. However, in this work, we are leaving open for future investigation issues such as ranking pages within Directories, users' perception of our system's performance, etc. It is our hope though, that our approach, will road the map for future improvements in populating Web Directories and in handling the proliferating Web data.

We now discuss a number of advantages that our approach entails and which we believe could be fruitfully explored by others. The implications of our findings apply primarily to Web cataloguers and catalogue users. Since cataloguers are challenged by the prodigious volume of the Web data that they need to process and categorize into topics, it is of paramount importance that they are equipped with a system that carries out on their behalf a preliminary categorization of pages. We do not imply that humans do not have a critical role to play in Directories' population, but we deem their "sine-qua-non" involvement in the evaluation and improvement of the automatically produced categorizations, rather than in the scanning of the numerous pages enqueued for categorization. In essence, we argue that our approach compensates for the rapidly evolving Web, by offering Web cataloguers a preliminary categorization for the pages that they have not processed yet. On the other side of the spectrum, end users are expected to benefit from the Directories' updated content. Given that users get frustrated when they encounter outdated pages every time they access Web catalogs to find new information that interests them, it is vital that Directories' contents are up-to-date. Our model ensures that this requirement is fulfilled, since it runs fast and scales up with the evolving Web, enabling immediacy of new data.

References

1. MultiWordNet Domains http://wndomains.itc.it/.
2. Open Directory Project http://dmoz.com.
3. Sumo Ontology http://ontology.teknowledge.com/.
4. WordNet 2.0 http://www.cogsci.princeton.edu/~wn/.
5. Apte C., Damerau F. and Weiss S.M. 1994. Automated learning of decision rules for text categorization. In *ACM Transactions on Inf. Systems*, 12(3):233-251.
6. Barzilay R. and Elhadad M. 1997. Lexical chains for text summarization. Master's Thesis.
7. Boyapati V. 2002. Improving text classification using unlabeled data. In Proceedings of *SIGIR Conference*, 11-15.
8. Broder A.Z., Glassman S.C., Manasse M. and Zweig G. 1997. Syntactic clustering of the web. In Proceedings of the *6th WWW Conference*: 1157-1166.
9. Chakrabarti S., Dom B., Agraval R. and Raghavan P. 1998(a). Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. In *VLDB Journal*, 7: 163-178.
10. Chakrabarti S., Dom B. and Indyk P. 1998(b). Enhanced hypertext categorization using hyperlinks. In Proceedings of *ACM SIGMOD* Conference.
11. Stamou S., Krikos V., Kokosis P. and Christodoulakis D. 2005. Web directory construction using lexical chains. In *Proceedings of the 10th NLDB Conference*.
12. Chen H. and Dumais S. 2000. Bringing order to the web: Automatically categorizing search results. In Proceedings of the *SIGCHI Conference*: 145-152.
13. Christianini N. and Shawe-Taylor J. 2000. *An introduction to support vector machines*. Cambridge University Press.
14. Duda R.O and Hart P.E.1973. *Pattern Classification and scene analysis*. Wiley & sons.
15. Furnkranz J. 1999. Exploring structural information for text classification on the WWW. In *Intelligent Data Analysis*: 487-498.
16. Glover E., Tsioutsiouliklis K., Lawrence S., Pennock M. and Flake G. 2002. Using web structure for classifying and describing Web pages. In Proc. of the *11th WWW Conference*.
17. Huang C.C., Chuang S.L. and Chien L.K. 2004. LiveClassifier: Creating hierarchical text classifiers through Web corpora. In Proceedings of the *13th WWW Conference*: 184-192.

18. Kaufman L. and Rousseeuw P.J. 1990. *Finding groups in data: An introduction to cluster analysis*. New York: Wiley & sons.
19. Koller D. and Sahami M. 1997. Hierarchically classifying documents using very few words. In Proceedings of *ICML Conference*: 170-178.
20. Mladenic D. 1998. Turning Yahoo into an automatic web page classifier. In the *13th European Conference on Artificial Intelligence*: 473-474.
21. Nigam K., McCallum A.K., Thrun S. and Mitchell T.M. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3): 103-134.
22. Ntoulas A., Cho J. and Olston Ch. 2004. What's new on the Web? The evolution of the Web from a search engine perspective. In Proceedings of the *13th WWW Conference*: 1-12.
23. Page L., Brin S., Motwani R. and Winograd T. 1998. The pagerank citation ranking: Bringing order to the web. (http://dbpubs.stanford.edu:8090/pub/1999-66).
24. Pazzani M. and Billsus D. 1997. Learning and revising user profiles: The identification of interesting Web sites. In *Machine Learning Journal*, 23: 313-331.
25. Resnik Ph. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. In *Journal of Artificial Intelligence Research*, 11: 95-130.
26. Ruiz M.E. and Srinivasan P. 1999. Hierarchical neural networks for text categorization. In Proceedings of *SIGIR Conference*: 281-282.
27. Song Y.I., Han K.S. and Rim H.C. 2004. A term weighting method based on lexical chain for automatic summarization. In Proceedings of the *5th CICLing Conference*: 636-639.
28. Krikos V., Stamou S., Ntoulas A., Kokosis P. and Christodoulakis D. 2005. DirectoryRank: ordering pages in web directories. In Proceedings of the 7th ACM International Workshop on Web Information and Data Management (WIDM), Bremen, Germany.
29. WordNets in the world. Available at http://www.globalwordnet.org/gwa/wordnet_table.htm