Web Directory Construction Using Lexical Chains

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

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

Abstract. Web Directories provide a way of locating relevant 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 present a way for automating the creation of a Web Directory. At a high level, our method takes as input a subject hierarchy and a collection of pages. We first leverage a variety of lexical resources from the Natural Language Processing community to enrich our hierarchy. After that, we process the pages and identify sequences of important terms, which are referred to as lexical chains. Finally, we use the lexical chains in order to decide where in the enriched subject hierarchy we should assign every page. Our experimental results with real Web data show that our method is quite promising into assisting humans during page categorization.

1 Introduction

Millions of users today access the plentiful Web content to locate information that is of interest to them. However, the task of locating relevant information is becoming daunting as the Web grows larger. Currently, there are two predominant approaches that users follow in order to satisfy their information needs on the Web: searching and browsing [25]. During searching, the users visit a Web Search Engine (e.g. Google) and use an interface to specify a query which best describes what they are looking for. During browsing, the users visit a Web Directory (e.g. the Yahoo! Directory), which maintains the Web organized in subject hierarchies, and navigate through these hierarchies in the hope of locating the relevant information. The appearance of a variety of Web Directories in the last few years (such as the Yahoo! Directory [8], the Open Directory Project (ODP) [4], the Google Directory [1] etc.) indicates that the Web Directories are a popular means for locating information on the Web.

Typically, the information provided by a Web Search Engine is automatically collected from the Web without significant human intervention. However, the construction and maintenance of a Web Directory involves a staggering amount of human effort because it is necessary to assign an accurate subject to every page inside the Web Directory. To illustrate the size of the effort necessary, one can simply consider the fact that Dmoz, one of the largest Web Directories, relies on more than 65,000

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 138–149, 2005.

volunteers around the world to locate and incorporate relevant information in the Directory. Given a Web page, one or more volunteers need to read it and understand its subject, and then examine Dmoz's existing Web Directory containing more than 590,000 subjects to find the best fit for the page. Clearly, if we could help the volunteers automate their tasks we would save a lot of time for a number of people.

One way to go about automating the volunteers' task of categorizing pages is to consider it as a classification problem. That is, given an existing hierarchy of subjects (say the Dmoz existing hierarchy) and a number of pages, we can use one of the many machine learning techniques to build a classifier which can potentially assign a subject to every Web page. One problem with this approach however, is that in general it requires a training set. That is, in order to build an effective classifier we need to first train it on a set of pages which has already been marked with a subject from the hierarchy. Typically this is not a big inconvenience if the collection that we need to classify and the hierarchy are static. As a matter of fact, as shown in [13, 15, 19, 22], this approach can be quite effective. However, in a practical situation, neither the Web [24] nor the subject hierarchies are static[1]. Therefore, in the case of the changing Web and subject hierarchy, one would need to recreate the training set and re-train the classifier every time a change was made.

In this paper, we present a novel approach for constructing a Web Directory which does not require a training set of pages and therefore can cope very easily with changes on the Web or the subject hierarchy. Our goal reaches beyond classification per se, and focuses on providing the means via which our hierarchy-based categorization model could be convenient in terms of both time and effort on behalf of Web cataloguers during page categorization. The only input that our method requires is the subject hierarchy from a Web Directory that one would like to use and the Web pages that one would like to assign to the Directory. At a very high level our method proceeds as follows: First we enrich the subject hierarchy of the Web Directory by leveraging a variety of resources created by the Natural Language Processing community and which are freely available. This process is discussed in Section 2. Then, we process the pages one by one and we identify the most important terms inside every page and we link them together, creating "lexical chains" which we will describe in Section 3. Finally, we use the enriched hierarchy and the lexical chains to compute one or more subjects to assign to every page, as shown in Section 4. After applying our method on a real Web Directory's hierarchy and a set of 114,000 Web pages we conclude that, in certain cases, our method has an accuracy of 87% into automatically assigning the Web pages to the same category that was selected by a human. Our results are presented in Section 5 and we conclude our work in Sections 6 and 7.

2 Building a Subject Hierarchy for the Web

Form a Web directory perspective, a subject hierarchy organizes a list of subjects, referred to as concepts, in successive ranks with the broadest listed first and with

[1] To see this one can simply compare Dmoz's subject hierarchies (file name: structure.rdf.u8.gz) in http://rdf.dmoz.org/rdf/archive

more specific aspects or subdivisions listed below. Typically, a hierarchy's concepts are depicted as nodes on a graph and the relations between concepts as arcs. Figure 1 shows a fraction of a hierarchy for the subject *Arts*, represented as a directed acyclic graph, where each node denotes a concept that is interconnected to other concepts via a specialization ("is-a") relation, represented by the dashed arcs. Concepts that are associated with a single parent concept are considered disjoint.

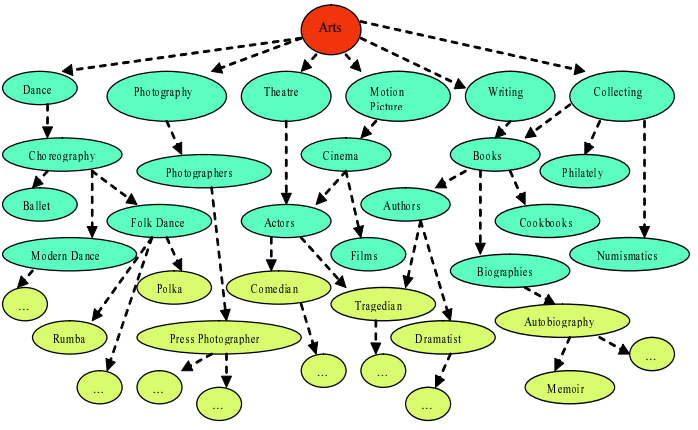

Fig. 1. A portion of the hierarchy for the *Arts* topic category (upper level topic) and subcategories (middle level concepts). The lower level nodes correspond to WordNet concepts.

For our purpose of generating a Web directory, we chose to develop a hierarchy of topics that are currently used by Web cataloguers in order to categorize Web pages thematically. To ensure that our hierarchy would both define concepts that are representative of the Web's topical content and be of good quality, we borrowed the hierarchy's top level concepts from the topic categories of the Google Directory and we enriched them with more specific concepts that we leveraged from existing ontological resources that have proved to be richly encoded and useful. The resources that we used for building our hierarchy are: (i) **The Suggested Upper Merged Ontology** (SUMO) [5]. SUMO is a generic ontology of more than 1,000 domain concepts that have been mapped to every WordNet synset that is related to them, (ii) **WordNet 2.0** [7]. WordNet is a lexical network of more than 118K synonym sets (synsets) that are linked to other synsets on the basis of their semantic properties and/or features, (iii) **MultiWordNet Domains** (MWND) [3]. MWND is an augmented version of WordNet; a resource that assigns every WordNet[2] synset a domain label among the total set of 165 hierarchically structured domains it consists of. Part of our hierarchy is illustrated in Figure 1. Our hierarchy has three different layers: the top layer corresponds to topics (Arts in our case); the middle layer to subtopics (e.g. Photography, Dance etc.) and the lower level corresponds to WordNet hierarchies. Our hierarchy can be downloaded from ftp://150.140.4.154/ftproot/.

[2] MWND labels were originally assigned to WordNet 1.6 synsets, but we augmented them to WordNet 2.0 using the mappings from http://www.cogsci.princeton.edu/~wn/links.shtml

2.1 Defining the Hierarchy's Concepts

To build our hierarchy we firstly enriched WordNet 2.0 with domain information. Note that a portion of WordNet 2.0 synsets are annotated with domain labels. For assigning domain labels to the remaining synsets, we leveraged domain knowledge from SUMO and MWND. To that end, we automatically appended (using available mappings) to every WordNet synset its corresponding domain label taken from either SUMO or MWND. Synsets of multiple domain assignments (i.e. synsets appended with a SUMO domain and a different MWND domain) were examined in order to select the domain that would best represent the synsets' semantics from a text categorization perspective. In selecting the most representative domain label among multiple domains, we adopted the Specification Marks technique, described in [27], which proceeds as follows. Given a group of terms that pertain to a domain category, we retrieve all their senses and supply them to a WSD module, which disambiguates them on the basis of the concept that is common to all the senses of all the words in this group. Disambiguated words are then supplied to a rules module, which locates the main-concept in WordNet for each of the domains, by using the hyper/hyponymy relation. The domain name used to label the main concept within a group of concepts is selected and propagated down to the WordNet terms that subsume the ISM concept, by following the hyponymy links. Finally, we manually examined the assigned domain labels and corrected any inconsistencies caused by erroneous disambiguation.

Having assigned a domain label to each WordNet synset, the next step we took was to define the hierarchy's top level topics. The hierarchy' top level concepts were chosen manually and they represent topics employed by Web cataloguers to categorize pages by subject. In selecting the topical categories we operated based on the requirement that our topics should be popular (or else useful) among the Web users. To that end, we borrowed 6 first level topics from the Google Directory taxonomy, thus satisfying our popularity requirement. These topics formed the hierarchy's root concepts and are the following: **Sports**, **Society**, **Sciences**, **Health**, **Arts** and **Recreation**. Following on, we integrated the WordNet hierarchies that have been enriched with domain information to their corresponding top level topics. Incorporating WordNet hierarchies into the six top level topics was carried out manually following an iterative process. The first straightforward step that we took was to select those WordNet hierarchies whose parent concept was labeled with a domain name identical to a top level topic (borrowed from the Google Directory) and automatically append them to the respective topic. The remaining hierarchies were manually appended to the hierarchy's top level topics based on their WordNet hypernymy relation. In particular, the WordNet hierarchies whose elements subsumed a top level topic were appended to this topic via an "is-a" relation. This way, those hierarchies' parent nodes become sub-domains in their corresponding 6 topics, denoting a middle level concept in the hierarchy. Following the steps described above, we integrated in our hierarchy's top level topics all WordNet lexical hierarchies for which a matching topic was found. At the end of the merging process, we came down to a total set of 143 middle level concepts, which were manually linked to the 6 top level topics, using their respective WordNet relations. The resulting upper level hierarchy (i.e. top and middle

level concepts) is a directed acyclic graph with maximum depth 4 and maximum branching factor, 28.

3 Reducing Pages to Lexical Chains

In this section we show how to leverage our hierarchy in order to detect which of the Web page's words are informative of the page's topic. We start our discussion by making the assumption that we process only Web pages written in English. Having automatically downloaded Web pages, we parse them to remove HTML markup, and we process the pages' contents by applying tokenization, stemming and part-of-speech tagging. Following, we eliminate stop-words from the pages and we compute a set of indexing keywords for every Web page. A driving factor in keywords' selection is to choose terms that express the pages' thematic content. Consequently, we need to account for the pages' lexical cohesion, i.e. the semantic relations that hold between the pages' terms. In selecting keywords, we augment the lexical chaining method introduced in [11, 18, 23], and for every Web page we automatically generate sequences of thematic words, i.e. sequences of semantically related terms.

The computational model we adopted for generating lexical chains is presented in the work of Barzilay [11] and it generates lexical chains in a three steps approach: (i) select a set of candidate terms[3] 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 and update accordingly. The relatedness factor in the second step is determined by the type of the links that are used in WordNet for connecting the candidate term to the terms that are already stored in existing lexical chains. Barzilay introduces also a "greedy" disambiguation algorithm that constructs all possible interpretations of the source text, using lexical chains.

However, Soung et al. [26] noted some caveats in this disambiguation formula in avoiding errors, because it does not consider anything about words that make a semantic relation. To surpass this limitation, they propose a new disambiguation formula, which relies on a scoring function *f*, which indicates a 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*, is calculated as the product of the words' association score, their depth in WordNet and their respective relation weight. The association score (*Assoc*) of the word pairs (w_1, w_2) is determined by the words' co-occurrence frequency in a corpus given by:

$$Assoc(w_1, w_2) = \frac{\log (p(w_1, w_2) + 1)}{N_s(w_1) \bullet N_s(w_2)} \mathrm{x} + \mathrm{y} = \mathrm{z} \quad (1)$$

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 senses that a word w has. The words' (w_1, w_2) depth (*DepthScore*) expresses the words' position in WordNet hierarchy and demonstrates that the lower a word is in WordNet hierarchy, the more specific meaning it has. Depth score is defined as:

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

$$DepthScore(w_1, w_2) = Depth(w_1)^2 \bullet Depth(w_2)^2 \quad (2)$$

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

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

The score of 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:

$$Score(C_i) = \sum_{r_j\ in\ C_j} f_s\ (w_{j1}, w_{j2}, r_j) \quad (4)$$

To compute a single lexical chain for every downloaded Web page, we segment the latter into shingles, using the shingling technique, described in [12]. To form a shingle, we group *n* adjacent words of a page, with $n = 50$, which roughly corresponds to the number of words in a typical paragraph. For every shingle, we generate and score lexical chains using the formula described above. In case a shingle produces multiple lexical chains, the chain of the highest score is regarded as the most representative shingle's chain, eliminating hence 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. This way we reassure that the overall score of every page's lexical chain is maximal. The elements of each chain are used as keywords for indexing the underlying pages in subject directories. In the subsequent paragraphs, we introduce a model that automatically categorizes Web pages into topics.

4 Assigning Web Pages to Topic Directories

In detecting the Web pages' topics, our model maps the pages' thematic keywords to the hierarchy's concepts, and traverses the hierarchy's matching nodes up to the root nodes. Recall that thematic words are disambiguated upon lexical chains' generation, ensuring that every keyword is mapped to a single node in the hierarchy. Traversal of the hierarchy's nodes accounts to following the hypermymic links of every matching concept until all their corresponding root topics are retrieved. For short documents with very narrow subjects there might be a single matching topic. However, due to both the sparseness of the Web data and the richness of our hierarchy, it is often the case that pages' thematic words correspond to multiple root topics. To accommodate multiple topics assignment, we compute a *Relatedness Score* (*RScore*) of every Web page to each of the hierarchy's matching topics. This relatedness score indicates how expressive is each of the hierarchy's topics for describing the Web pages' contents. Formally, the relatedness score 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 the category D_k. The *Relatedness Score* that a page has to each of the hierarchy's matching topics is given by:

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

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 zero and one, 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 indexed in the topical category D_k for which it has the highest relatedness score of all its *RScores* above a threshold T, for T = 0.5. The page's indexing score is:

$$IScore\ (\mathrm{i}, \mathrm{k}) = \max RScore\ (\mathrm{i}, \mathrm{k}) \quad \text{where } 1 \le i \le T \tag{6}$$

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

5 Experimental Study

To study the efficiency of our approach in populating Web directories, we conducted an experiment in which we supplied our model, named TODE, with 114K Web pages, inquiring that these are categorized in the appropriate topics in the hierarchy.

To ascertain that our perception of TODE's performance would not entail any bias, we elected to experiment with Web pages that had already been listed in topical categories by domain experts. In selecting the experimental data, we downloaded pages from those topics in Google Directory that matched any of the topics represented in our hierarchy, i.e. top level concepts. Downloading took a few days and we fetched only the pages that had been explicitly assigned to one of the six topics in Google Directory, without following the pages' internal links. By selecting pages from the Google Directory, we believe that our sample was popular and of good quality, which is implied by the large number of Google Directory users. In total, we downloaded 114,358 pages that span 91 Google Directory second level topics, which in turn are organized into 6 first level topics. Recall that the 6 first level topics in the Google Directory are the same with the top level topics in our hierarchy. The size of the downloaded data is nearly 5.9GB, which is reduced to 638MB when compressed. Table 1 shows the fraction of experimental pages in each topic in Google Directory.

Table 1. Distribution of Google Directory Topics in our Data.

Topics	Fraction of pages in topic
Society	29%
Arts	25%
Sciences	25%
Recreation	10%
Sports	9%
Health	2%

We parsed the downloaded pages and generated the shingles for them after removing the HTML markup. Pages were then tokenized, tagged, lemmatized and submit-

ted to TODE, which computed and weighted a single lexical chain for every page. To compute lexical chains, our model relied on a resources index which comprised: (i) the 12.6MB WordNet 2.0 data, (ii) a 0.5GB compressed corpus, which we processed in order to obtain statistical data about words' co-occurrence frequencies, and (iii) the 11MB upper level topics and subtopics in our hierarchy. Using the above data, TODE generated and scored lexical chains for every page; it computed the pages' *RScores* and *IScores* and stored this information in a secondary index. Based on a combined analysis of the data stored in the secondary index, TODE indicates the most appropriate hierarchy's (sub)-topic(s) to categorize each of the pages. At the end of the experiment, we compared the categorizations given by TODE for each of the pages to the categorizations these pages had in the respective Google Directory categories. Our comparison revealed that our model assigned 88,237 out of the 114,358 pages to a category and failed to deliver categorizations for the remaining 26,121 pages. Categorization failure was due to: (i) lack of textual data in the underlying pages; pages comprised lists of links, audiovisual data, etc., (ii) non-existent pages; dead links, redirects, (iii) frames in pages, (iv) downloading time-outs after 10 seconds, and (v) inefficiency in generating lexical chains for pages with very short content. The results presented below are based on the categorizations given for the 88,237 pages.

5.1 Directories' Population Performance

To evaluate our system's performance, we used as a comparison testbed the categorizations that our experimental data displayed in the respective Google Directory (sub)-topics. This is primarily because in Google Directory, pages have been manually assigned to topical categories, and secondly because of Google Directory's popularity, which stipulates that the offered categorizations have been found to be useful by many people. We first report the overall performance of our system in categorizing pages into topics, by comparing the fraction of pages that have been assigned to the same topics in both our hierarchy and Google Directory. Figure 2 plots the results.

In Figure 2, the horizontal axis represents the six top-level topics that are common between our hierarchy and the Google Directory. The vertical axis shows the fraction of pages that have been assigned to each of the topics, where 100% corresponds to the total number of pages for which TODE delivered categorizations, i.e. the 88,237 pages. For every topic, the solid/gray bars represent the fraction of pages categorized in that topic in Google Directory and the decorated bars represent the fraction of pages that have been categorized to that topic by TODE. The dark colored part of the decorated bars corresponds to the fraction of TODE's successful categorizations, i.e. the fraction of pages that have been assigned to the same topic by our system as in Google Directory, whereas the light colored part of the decorated bars corresponds to the fraction of pages mis-categorized by TODE. As mis-categorizations, we consider the pages that have been assigned by TODE to a topic, but which they are not assigned to the same topic in Google Directory. From the graph, we can see that in overall our system has a satisfactory performance in detecting a dominant topic of the Web pages, considering that the entire process was fully automated.

In order to have a clearer view on the obtained results, we give percentage values of the delivered categorizations in Table 2, whose first column shows the topics used in our experiment, the second column shows the fraction of the experimental pages that were assigned to each topic in Google Directory, the third column shows the fraction of pages that have been assigned to each topic in TODE and the forth column shows the fraction of the pages that have been "successfully" assigned to each of the topics in TODE, in the sense that these pages are also categorized in the same topics in Google Directory. By quantifying the amount of "successful" categorizations for all of the six topics together, we can see that our system had on average 75.1% categorization accuracy. Note that, categorization accuracy is defined as the fraction of the 88,237 pages that have been assigned to the same topic in our model as in Google Directory. Experimental results, verify that our system has a noticeable potential in assigning pages to topical categories, without imposing any need for human intervention, nor requiring training. Based on our experimental findings, we argue that our system could be employed as a Web cataloguers' assistant and deliver preliminary categorizations for Web pages. These preliminary categorizations could be then further examined by human cataloguers and reordered when necessary.

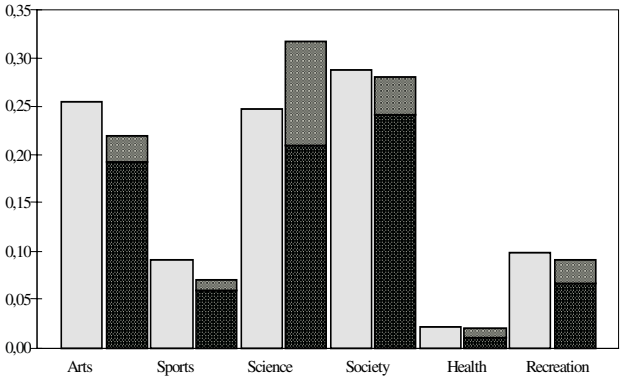

Fig. 2. Fraction of pages assigned to topics in Google Directory (solid/gray bars) and in TODE (decorated bars). Dark parts of the decorated bars correspond to the fraction of pages assigned to the same topic in both Google Directory and TODE while light colored parts correspond to the fraction of pages assigned to that topic in TODE but in another topic in Google Directory.

Table 2. Categorization Results.

Topics	Pages in Google Directory	Pages in TODE	TODE pages compatible with Google Directory
Society	29%	28%	86%
Arts	25%	22%	87%
Sciences	25%	32%	66%
Recreation	10%	9%	73%
Sports	9%	7%	85%
Health	2%	2%	54%

Our next experiment considered the use of the Google Directory hierarchies for categorizing the same 88,237 pages into the ontology's 6 top level topics. In particular, we appended to each of the ontology's 6 upper level topics their corresponding second and third level concepts in the Google Directory hierarchy, and we enriched the resulting ontology with lower level concepts that we borrowed from WordNet and which correspond to the hyponyms of the Google Directory concepts. We then incorporated this new ontology in TODE and we supplied our system with the 88,237 pages of our first experiment inquiring that these are categorized in the same 6 top level topics. Obtained categorizations were again compared to the categorizations the experimental pages have in the Google Directory. Our results confirm the potential that TODE has in successfully finding a dominant topic for categorizing Web pages, which accounts to an average categorization accuracy of 73%. Although TODE's performance figures might seem somehow low at first glance, however we believe that they are quite promising if we consider that the entire process was fully automatic, without the need of any training or human intervention.

6 Related Work

There has been previous work in categorizing Web pages into pre-defined topics [13, 14, 16, 22]. Related work falls into four categories. The first one concerns the hierarchical organization of the Web pages that are retrieved by search queries [15, 21]. This could also be addressed from a meta-search engine perspective, which aims to cluster together the pages retrieved in response to a query, based on either the contents of the pages [6], their links' structure [2], or both [16, 17]. Studies falling into this category rely significantly on the issued queries and the pages retrieved as relevant to the queries at hand. Our work differs from these studies, because we are not dealing with a subset of pages already deemed as relevant to a query (i.e. topic) by some searching mechanism. Instead, we aim at organizing Web pages by relying exclusively on the pages' thematic words and their semantic correlations. The second category concerns the automatic grouping of Web pages into personalized Web directories, based on user-profiling techniques [9, 10]. These approaches employ document clustering methods and usage mining techniques, to automate the process of organizing Web pages into topics. But, these techniques rely on a relatively small and precise set of "interesting" topics that are supplied to the various classification schemes as training paradigms. These training paradigms are determined by the users themselves, either in an explicit manner, by informing the system on their preferences (profiling), or implicitly, by having the system learn the users' profiles from their previous navigational behavior. Our approach for categorizing Web pages by topic is far more generic than personalized classification, in the sense that it is not bound to any particular information preferences and does not undergo any training phase. The third category relies on text classification techniques that group Web pages into pre-existing topics [14, 19]. In this approach, statistical techniques are used to learn a model based on a labeled set of training documents. This model is then applied to unlabeled pages to determine their topics. Again, the distinctive feature of our model from text classification techniques is the lack of a training phase. Finally, the objec-

tive in our work (i.e. directories' population) could be addressed from the agglomerative clustering [20] perspective; a technique that treats the generated agglomerate clusters as a topical hierarchy for clustering documents. The agglomerative clustering methods build the subject hierarchy at the same time as they generate the clusters of the documents. In our work, we preferred to build our hierarchy by using existing resources, rather than to rely on newly generated clusters, for which we would not have enough evidence to support their usefulness for Web pages' categorization.

7 Conclusion

We have presented a model that explores 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 efficiency of our model in categorizing a fraction of Web pages into topical categories, and contrasted its resulting categorizations to the categorizations that the same pages displayed in the corresponding Google Directory categories. Our findings indicate that our model has a promising potential in facilitating current tendencies in editing and maintaining Web directories. 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.

References

1. Google Directory http://dir.google.com.
2. Kartoo http://www.kartoo.com.
3. MultiWordNet Domains http://wndomains.itc.it/.
4. Open Directory Project http://dmoz.com.
5. Sumo Ontology http://ontology.teknowledge.com/.
6. Vivisimo http://www.vivisimo.com/.
7. WordNet 2.0 http://www.cogsci.princeton.edu/~wn/.
8. Yahoo! http://yahoo.com.
9. Yahoo! Inc. *MyYahoo* http://my.yahoo.com
10. Anderson C.R. and Horvitz E. Web montage: A dynamic personalized start page. In Proceedings of the 11th WWW Conference, 2002, 704-712.
11. Barzilay R. and Elhadad M. Lexical chains for text summarization. Master's Thesis, Ben-Gurion University, 1997.
12. Broder A.Z., Glassman S.C., Manasse M. and Zweig G. Syntactic clustering of the web. In Proceedings of the 6th WWW Conference, 1997, 1157-1166.
13. Chakrabarti S., Dom B., Agraval R. and Raghavan P. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. VLDB Journal, *7*, 1998, 163-178.
14. Chekuri C., Goldwasser M., Raghavan P. and Upfal E. Web search using automated classification. In Proceedings of the 6th WWW Conference, 1997.

15. Chen H. and Dumais S. Bringing order to the web: Automatically categorizing search results. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2000), 145-152.
16. Halkidi M., Nguyen B., Varlamis I. and Vazirgiannis M. THESUS: Organizing web document collections based on link semantics. VLDB Journal, *12*, 2003, 320-332.
17. Haveliwala T. Topic sensitive PageRank. In Proceedings of the 11th WWW Conference, 2002, 517-526.
18. Hirst G. and St-Onge D. Lexical chains as representations of content for the detection and correction of malapropisms. In Fellbaum Ch. (ed.), WordNet: An Electronic Lexical Database. MIT Press, 1998, 305-332.
19. Huang C.C., Chuang S.L. and Chien L.K. LiveClassifier: Creating hierarchical text classifiers through web corpora. In Proceedings of the 13th WWW Conference, 2004, 184-192.
20. Kaufman L. and Rousseeuw P.J. Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & sons, 1990.
21. Kummumuru K., Lotlikar R., Roy S., Singai K. and Krishnapuram R. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In Proceedings of the 13th WWW Conference, 2004, 658-655.
22. Mladenic D. Turning Yahoo into an automatic web page classifier. In Proceedings of the 13th European Conference on Artificial Intelligence, 1998, 473-474.
23. Morris J. and Hirst G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17, 1, (1991), 21-43.
24. Ntoulas A., Cho J. and Olston Ch. What's new on the web? The evolution of the web from a search engine perspective. In Proceedings of the 13th WWW Conference, 2004, 1-12.
25. Olston Ch. and Chi E. ScentTrails: Intergrading browsing and searching. ACM Transactions on Computer-Human Interaction 10, 3 (Sept. 2003), 1-21.
26. Song Y.I., Han K.S. and Rim H.C. A term weighting method based on lexical chain for automatic summarization. In Proc. of the 5th CICLing Conference, 2004, 636-639.
27. Montoyo, A., Palomar, M., and Rigau, G. WordNet Enrichment with Classification Systems. In Proc. Of NAACL Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customization, 2001.