Cover Page

Paper Title:

TODE: An Ontology-Based Model for the Dynamic Population of Web Directories

Authors

Sofia Stamou

Computer Engineering and Informatics Department, Patras University, 26500, Greece

Phone: +30 2610 996 994

Fax: +30 2610 969 013

E-mail: stamou@ceid.upatras.gr

Alexandros Ntoulas

Computer Science Department, University of California Los Angeles (UCLA), CA 90095, USA

Phone: +1 310 795 4962

E-mail: ntoulas@cs.ucla.edu

Dimitris Christodoulakis

Computer Engineering and Informatics Department, Patras University, 26500, Greece

Phone: +30 2610 996 921

Fax: +30 2610 969 013

E-mail: dxri@upatras.gr

List of Keywords

Data processing

Information Organization

Web resources

Web-based Directory

Data Management

Knowledge Classification

TODE: An Ontology-Based Model for the Dynamic Population of Web Directories

ABSTRACT

*In this paper we study how we can organize the continuously proliferating Web content into topical categories, also known as Web directories. In this respect, we have implemented a system, named TODE that uses a **T**opical **O**ntology for **D**irectories' **E**diting. First, we describe the process for building our ontology of Web topics, which are treated in TODE as directories' topics. Then, we present how TODE interacts with the ontology in order to categorize Web pages into the ontology's topics and we experimentally study our system's efficiency in grouping Web pages thematically. We evaluate TODE's performance by comparing its resulting categorization for a number of pages to the categorization the same pages display in Google Directory as well as to the categorizations delivered for the same set of pages and topics by a Bayesian classifier. Results indicate that our model has a noticeable potential in reducing the human-effort overheads associated with populating Web directories. Furthermore, experimental results imply that the use of a rich topical ontology increases significantly classification accuracy for dynamic contents.*

INTRODUCTION

Millions of users today access the plentiful Web content to locate information that is of interest to them. However, as the Web grows larger the task of locating relevant information within a huge network of data sources is becoming daunting. Currently, there are two predominant approaches for finding information on the Web, namely searching and browsing (Olston and Chi, 2003). In the process of searching users visit a Web Search Engine (e.g. Google) and specify a query which best describes what they are looking for. During browsing, 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 construction of a variety of Web Directories in the last few years (such as the Yahoo! Directory (http://yahoo.com), the Open Directory Project (ODP) (http://dmoz.org), the Google Directory (http://dir.google.com) etc.) indicates that Web Directories have gained popularity as means for locating information on the Web.

Typically, the information provided by a Web Search Engine is automatically collected from the Web without any 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 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 of 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' tasks 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 both the collection that we need to classify and the hierarchy are static. As a matter of fact, as shown in (Chakrabarti et al., 1998a; Chen and Dumais, 2000; Huang et al., 2004; Mladenic, 1998), this approach can be quite effective. However, in a practical situation, neither the Web nor the subject hierarchies are static. For example, previous studies have shown that 8% of new pages show up on the Web every week (Ntoulas et al., 2004) and Dmoz's subject hierarchy is undergoing a variety of changes every month[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. 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 320,000 Web pages we conclude that, in certain cases, our method has an accuracy of 90.70% into automatically assigning the Web pages to the same category that was selected by a human. Our experimental results are presented in Section 5.

In summary, we believe that our work makes the following contributions:

- **Untangling the Web via an ontology**: We introduce an ontology[2] that has been designed to serve as a reference guide for grouping Web pages into topical categories. In particular, we report on the distinct knowledge bases that have been merged together to form the ontology. The resulting joint ontology was further augmented with a top level of topics, which are borrowed from the Google Directory subject hierarchy. We explore the ontology's lexical hierarchies to compute chains of thematic words for the Web pages. Dealing with lexical chains rather than full content, reduces significantly both the categorization process overhead and the computational effort of comparing pages, as we will shown in Section 4.
- **Bringing order to directories' contents**: We use the ontology to deliver a comprehensive ordering of Web pages into directories and to prune directories' overpopulation. In particular, we introduce DirectoryRank, a metric that sorts the pages assigned to each directory in terms of both their relatedness to the directory's topic and their correlation to other "important" pages grouped in the same directory.
- **Keeping up with the evolving Web**: The immense size of the Web is prohibitive for thoroughly investigating the information sources that exist out there. Our model enables the incremental editing of Web directories and can efficiently cope with the evolving Web. The efficiency of our system is well supported by empirical evidence, which proves that it gives good results and scales well. Therefore, directories remain "fresh" upon index updates and newly downloaded pages are accessible through Web catalogs, almost readily.

To the best of our knowledge our study is the first to make explicit use of an ontology of Web-related topics to dynamically assign Web pages to directories. Our goal reaches beyond classification per se, and focuses on providing the means via which our ontology-based categorization model could be convenient in terms of both time and effort on behalf of Web cataloguers in categorizing pages. In particular, we show that our approach can serve as a good alternative to today's practices in populating Web Directories.

We start our discussion by presenting how to enrich an existing subject hierarchy with information from the Suggested Upper Merged Ontology (http://ontology.teknowledge.com), WordNet (http://www.cogsco.princeton.edu/~wn) and MultiWordNet Domains (http://wndomains.itc.it). Construction of lexical chains is presented in Section 3, while Section 4 shows how to employ the lexical chains to assign Web pages to the subject hierarchy. Our experimental results are shown in Section 5 and we conclude our work in Sections 6 and 7.

BUILDING AN ONTOLOGY FOR THE WEB

Traditionally, ontologies are built in order to represent generic knowledge about a target world (Bunge, 1977). An ontology defines a set of representational terms, referred to as concepts, which describe abstract ideas in the target world and which can be related to each other. For example, in an ontology representing all living creatures, "human" and "mammal" might be two of the concepts and these two concepts might be connected with a relation "is-a" (i.e. human "is-a" mammal). Typically, ontologies' concepts are depicted as nodes on a graph and the relations between concepts as arcs. For example, **Figure 1** shows a fraction of an ontology for the topic *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 via an "is-a" link are considered disjoint.

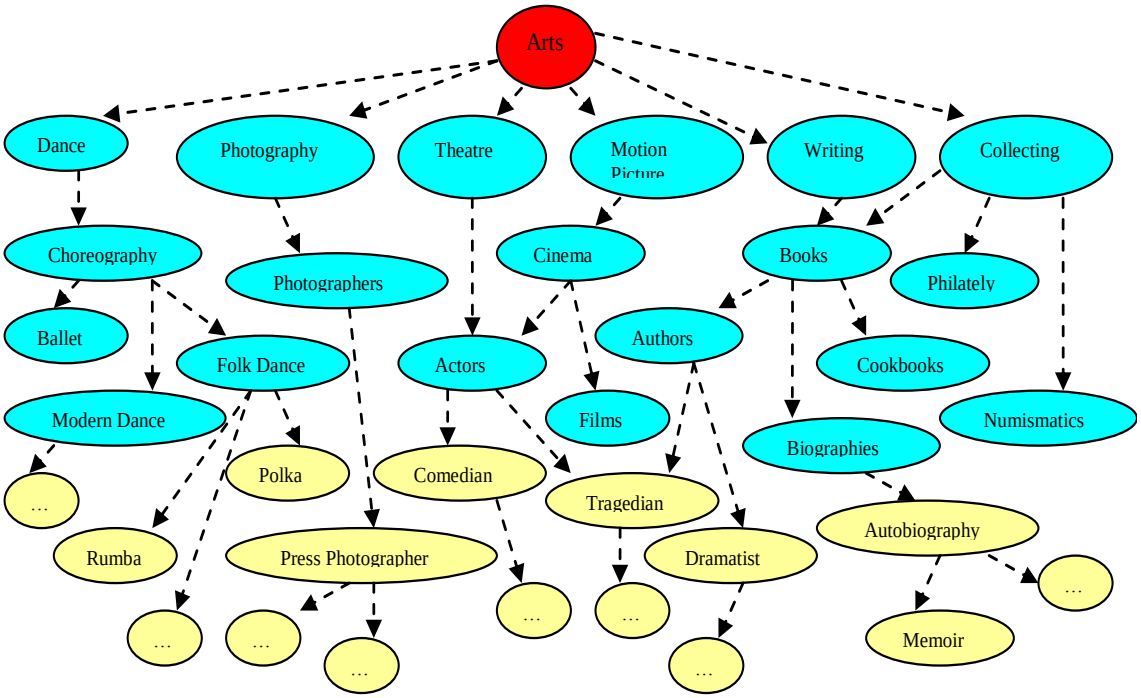

Figure 1. A portion of the ontology for the *Arts* topic category.

Depending on the application, there are different ways of developing an ontology. The usefulness, however, of an ontology lies in the fact that it presents knowledge in a way easy to understand by humans. For our purpose of generating a Web directory, we chose to develop an ontology that would describe humans' perception of the most popular topics that are communicated via the Web. Consequently, we define our ontology as a hierarchy of topics that are currently used by Web cataloguers in order to categorize Web pages in topics. To ensure that our ontology would define concepts that are representative of the Web's topical content, we borrowed the ontology's top level concepts from the topic categories of the Google Directory. Moreover, to guarantee that our ontology would be of good quality, we preferred to obtain our ontology's conceptual hierarchies from existing ontological resources that have proved to be richly encoded and useful. In order to build our ontology we used three different sources:

1. The Suggested Upper Merged Ontology (SUMO). 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.
2. WordNet 2.0. 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.
3. The MultiWordNet Domains (MWND). MWND is an augmented version of WordNet; a resource that assigns every WordNet[3] synset a domain label among the total set of 165 hierarchically structured domains it consists of.

The reason for using the above resources to build our ontology is the fact that they have been proven to be useful in resolving sense ambiguities, which is crucial in text categorization. Additionally, because the above resources are mapped to WordNet, our task of merging them into a common ontology is easier. Part of our ontology is illustrated in Figure 1. Our ontology has three different layers: the top layer corresponds to topics (Arts in our case), the middle layer to subtopics (for example Photography, Dance etc.) and the lower level corresponds to WordNet hierarchies, whose elements are hyponyms of the middle level concepts. We describe the selection of the topics in every layer next.

The Top Level Topics

The ontology' 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 following dual requirement: (i) our topics should be popular (or else useful) among the Web users and (ii) they should be sufficiently represented within WordNet, in order to guarantee that our ontology would be rich in concept hierarchies. To that end, we borrowed topics from the Google Directory taxonomy, thus satisfying our popularity requirement. Subsequently, we manually checked the topics against WordNet hierarchies and all Google Directory topic concepts found in WordNet and which had deep and dense subordinate hierarchies were retained, thus fulfilling the WordNet representation requirement. Eventually, we came down to totally 13 Google Directory first level topics, for which there was sufficient information within the WordNet hierarchies. These topics formed the ontology's root concepts and are shown on Table 1.

Table 1. The Ontology's Root Concepts

First Level Topics	
Arts	News
Sports	Society
Games	Computers
Home	Reference
Shopping	Recreation
Business	Science
Health	

The Middle and Lower Level Concepts

Middle level concepts were determined by merging MWND and SUMO into a single combined resource. Merging SUMO hierarchies and MWND domains into a common ontology 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 (Resnik, 1999).

The parent concept of every merged hierarchy was then searched in the ontology's 13 top level topics (borrowed from the Google Directory) and if there was a matching found, this merged hierarchy was integrated with this top level concept For instance, consider the SUMO hierarchies of the domain "swimming" and the hierarchies that have been assigned the MWND domain "sport". Due to their hierarchies' overlapping elements in WordNet, "sport" and "swimming" were integrated in a common parent concept, i.e. "sport". Because this parent concept is also a top level topic (***Sports/ Athletics***), the merged hierarchies are assigned to the ontology's topic ***Sports/Athletics***.

If no matching was found between the merged hierarchy's parent concept and the ontology's topics, the direct hypernyms of the parent concept were retrieved from WordNet and searched within the ontology's 13 top level topics. If there was a matching found, the merged hierarchy was integrated with the top level topic via the "is-a" relation. This way the joint hierarchy's parent concept becomes a sub-domain in one of the ontology's 13 topics, and denotes a middle level concept in the ontology. As an example, consider the SUMO domain concept "computer program", whose corresponding hierarchies have been integrated with the hierarchies of the MWND domain "applied science". Following merging, the joint hierarchies' parent concept was searched in the ontology's top level concepts. Because this parent concept was not among the ontology's topics, its WordNet direct hypernyms were retrieved and searched in the ontology's topics. Among the hypernyms of the concept "applied science" is the concept "science", which is also a top level topic in the ontology. As such, the hierarchies merged into the "applied science" concept, were integrated into the ***Science*** topic, and their common parent concept becomes a middle level concept in the ontology.

Following the steps described above, we integrated in the ontology's top level topics all SUMO and MWND hierarchies for which there was sufficient evidence in WordNet to support our judgments for their merging. The hierarchies that remained disjoint at the end of this process were

disregarded from the ontology. Although, we could have examined more WordNet hypernyms (i.e. higher level concepts), in an attempt to find a common parent concept to merge the remaining SUMO and MWND disjoint hierarchies, we decided not to do so, in order to weed out too abstract concepts from the ontology's middle level concepts. Our decision was based on the intuition that the higher a concept is in a hierarchy, the greater the likelihood that it is a coarse grained concept that may lead to obscure distinctions about the pages' topics. At the end of the merging process, we came down to a total set of 489 middle level concepts, which were subsequently organized into the 13 top level topics, using their respective WordNet relations. The resulting upper level ontology (i.e., top and middle level concepts) is a directed acyclic graph with maximum depth 6 and branching factor, 28 (i.e. number of children concepts from a node). Finally, we anchored to each middle level concept all WordNet hierarchies that encounter a specialization link to any of the ontology's middle level concepts. The elements in WordNet hierarchies formed our ontology's lower level concepts.

REDUCING PAGES TO LEXICAL CHAINS

In this section we show how to leverage the ontology that we generated, in order to detect which of the Web pages' words are informative of the page's theme. At a high level, we explore the ontology's concepts while processing Web pages in order to find the pages' thematic words. This results into generating for every page a sequence of lexical elements, known as lexical chains. Lexical chains communicate the pages' thematic content and they will be used later on to determine the Web pages' topical categories.

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 adopted for generating lexical chains is presented in the work of Barzilay and Elhadad (1997) and it generates lexical chains in a three steps approach: (i) select a set of candidate terms[4] 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. Figure 2 illustrates an example of the lexical chain generated for a text containing the candidate terms: system, network, sensor, weapon, missile, surface and net. The subscript *si* denotes the id of the word's sense within WordNet[5].

Lexical chain

system $_{s6}$ → network $_{s4}$

system $_{s6}$ → sensor $_{s1}$

system $_{s6}$ → weapon $_{s2}$ → missile $_{s1}$

system $_{s6}$ → surface $_{s1}$ → net $_{s2}$

Figure 2. A lexical chain example.

Having generated lexical chains, we disambiguate the sense of the words inside every chain by employing the scoring function *f* introduced in (Song et al., 2004), which indicates the probability 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' co-occurrence frequency in a generic corpus that has been previously collected. In practice, the greater the association score between a word pair w_1 and w_2 is, the greater the likelihood that w_1 and w_2 refer to the same topic. Formally, the (*Assoc*) score of the word pair (w_1, w_2) is given by:

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

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 a word w has.

Given a word pair (w_1, w_2) their *DepthScore* expresses the words' position in WordNet hierarchy and is defined as:

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

where *Depth* (w) is the depth of word w in WordNet and indicates that the deeper a word is in the WordNet hierarchy, the more specific meaning it has.

Within the WordNet lexical network two words w_1 and w_2 are connected through one or more relations. For example the words computer and calculator are connected through a synonymy relation, while the words computer and server are connected through a hyponymy relation. In our framework, 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:

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

The value of the function *f* represents the probability that the relation type *r* is the correct one between words w_1 and w_2. In order to disambiguate the senses of the words within lexical chain C_i we calculate its score, by summing up the f_s scores of all the words w_{j1} w_{j2} (where w_{j1} and w_{j2} are successive words) within the chain C_i. Formally, the score of lexical chain C_i, is expressed as the sum of the score of each relation r_j in C_i.

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

Eventually, in order to disambiguate we will pick the relations and senses that maximize the *Score* (C_i) for that particular chain.

To compute a single lexical chain for every downloaded Web page, we segment the latter into shingles (Broader et al., 1997), 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 consecu-

tively. 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.

POPULATING WEB DIRECTORIES

We have so far described how Web pages are reduced into sequences of thematic words, which are utilized by our model for categorizing Web pages to the ontology's topics. Here, we analyze how our model (TODE) populates topic directories and we evaluate its efficiency in categorizing roughly 320,000 real Web pages.

Assigning Web Pages to Topic Directories

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:

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

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:

IScore (i, k) = max *RScore* (i, k)

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.

Ordering Web Pages in Topic 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:

$$AugElements(PC_i) = C_i \bigcup SynC_i \text{ and } AugElements(PC_j) = C_j \bigcup SynC_j$$

where, $SynC_i$ denotes the set of the ontology's concepts that are synonyms to any of the elements in C_i and $SynC_j$ denotes the set of the ontology'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:

$$ComElements(PC_i, PC_j) = AugElements_i \bigcap AugElements_j$$

We formally define the problem of computing pages' semantic similarities as follows: if pages P_i and P_j share elements in common, produce the correlation look up table with triples 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 computed as follows:

$$\sigma_S(PC_i, PC_j) = \frac{2 \bullet \left| ComElements \right|}{\left| AugElements_i \right| + \left| AugElements_j \right|}$$

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.

Ranking Pages in Directories

Pages are sorted in Directory topics on the basis of a DirectoryRank metric, which defines the importance of the pages with respect to the particular topics in the Directory. Note that in the context of Web Directories, we perceive the amount of information that a page communicates about some Directory topic to be indicative of the page's importance with respect to the given topic.

DirectoryRank (*DR*) measures the quality of a page in some topic by the degree to which the page correlates to other informative/qualitative pages in the given topic. Intuitively, an informative page in a 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 $p_1, p_2, \ldots, p_n$ 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:

$$DR\ (p_i, T_k) = RScore\ (p_i, T_k) + \left[\sigma_s\ (PC_1, PC_i) + \sigma_s\ (PC_2, PC_i) + \ldots\ldots + \sigma_s\ (PC_n, PC_i)\right] / n$$

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 the Directory, and that (ii) more users are likely to visit them while browsing the Directory's contents. Lastly, it should be noted that similarities are computed offline for all the pages in a Directory's topics, regardless of the pages' *RScore*.

EXPERIMENTAL STUDY

We have implemented the experimental TODE prototype using a Pentium 4 server at 2.4 GHz, with 512 MB of main memory. For fast computations of the lexical chains, we stored the ontology's top and middle level (sub)-topics in main memory, while WordNet hierarchies were stored on disk and were accessed through a hash-table whenever necessary. Moreover, words' co-occurrence statistics were pre-computed in the corpus and stored in inverted lists, which were again made accessible upon demand. Of course, the execution time of TODE's categorizations depends on both the number of pages considered and the ontology's coverage. In our experimental setup it took only a few hours to categorize our whole dataset. In order to study the efficiency of our approach in populating Web directories, we conducted an experiment in which we supplied TODE with roughly 320K Web pages, inquiring that these are categorized in the appropriate top or middle level ontology's concepts.

Experimental pages were obtained from the Google Directory, because of the Google Directory topics' compatibility with our ontology's topics. A less decisive factor for picking our data from the Google Directory is because the latter maintains a ranked list of the pages associated with each category. At the end of the experiment, we compared our model's resulting categorizations to the categorizations the same pages displayed for the same topics in Google Directory as well as to the categorizations delivered for the same set of pages and topics by a Naïve Bayes classifier. In this section, we present our experimental data and we discuss TODE's classification performance based on experimental results.

Experimental Data

In selecting our experimental data, we wanted to pick a useful yet representative sample of the Google Directory 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 Google Directory categories, whose topics are among the top level topics in our subject hierarchy.

In selecting our experimental data, we picked pages that are categorized in those topics in Google Directory, which are also present in our hierarchy. Recall that we borrowed our hierarchy's 13 top-level topics form Google Directory.

Out of all the sub-topics organized in those 13 top-level topics in Google Directory, 156 were represented in our hierarchy. Having determined the topics, whose set of pages would be categorized by our system, we downloaded a total number of 318,296 pages, categorized in one of the

156 selected topics, which in turn are organized into the 13 top-level topics. Table 2 shows the statistical distribution of our experimental pages in the selected top level topics in Google Directory.

Table 2. Statistics on the experimental data

Category	# of documents	# of sub-topics
Arts	28,342	18
Sports	20,662	26
Games	11,062	6
Home	6,262	7
Shopping	52,342	15
Business	60,982	7
Health	23,222	7
News	9,462	4
Society	28,662	14
Computers	35,382	13
Reference	13,712	10
Recreation	8,182	20
Science	20,022	9
Total	318,296	156

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.

Since we were interested in evaluating the performance of our approach in automatically categorizing web pages to the ontology's topics, 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 (Duda and Hart, 1973). In particular, we 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 Google Directory 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 Google Directory category. For evaluating the classification accuracy of both the Bayesian and our classifier, we used the Google Directory categorizations as a comparison testbed, i.e. we compared the classification delivered by each of the two classifiers to the classification done by the Google Directory cataloguers for the same set of pages. Although, our experimental pages are listed in all

sub-categories of the Google Directory' s top level topics, for the experiment presented here, we mainly focus on classifying the Web pages for the top-level topics.

Directories' Population Performance

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.

Table 3. **Overall accuracy results of 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%

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 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).

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 (Apte et al., 1994), Support Vector Machines (Christianini and Shawe-Taylor, 2000), Bayesian classifiers (Pazzani and Billsus, 1997), hierarchical text classifiers (Koller and Sahami, 1997; Stamou at al., 2005; Chakrabarti et al., 1998a; Mladenic, 1998; Ruiz and Srinivasan, 1999; Chen and Dumais, 2000; Nigam et al., 2000; Boypati, 2002; Huang et al., 2004). 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 (Chakrabarti et al., 1998b; Furnkranz, 1999; Glover et al., 2002). 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 (Kaufman and Rousseeuw, 1990). 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.

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. It is our hope therefore, 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

Google Directory http://dir.google.com.

MultiWordNet Domains http://wndomains.itc.it/.

Open Directory Project http://dmoz.com.

Sumo Ontology http://ontology.teknowledge.com/.

WordNet 2.0 http://www.cogsci.princeton.edu/~wn/.

Yahoo! http://yahoo.com.

Apte C., Damerau F. & Weiss S.M. (1994). Automated learning of decision rules for text categorization. In *ACM Transactions on Information Systems,* 12(3):233-251.

Barzilay R. & Elhadad M. (1997). *Lexical chains for text summarization*. Master's Thesis, Ben-Gurion University.

Boyapati V. (2002). Improving text classification using unlabeled data. In *Proceedings of the ACM Special Interest Group in Information Retrieval (SIGIR) Conference,* (pp. 11-15).

Broader A.Z., Glassman S.C., Manasse M. & Zweig G. (1997). Syntactic clustering of the web. In *Proceedings of the 6th International World Wide Wweb (WWW) Conference*, (pp.1157-1166).

Bunge M. (1977). *Treatise on Basic Philosophy. Ontology I. The Furniture of the World*. Vol. 3, Reidel, Boston.

Chakrabarti S., Dom B., Agraval R. & Raghavan P. (1998)a. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. In *Very Large DataBases (VLDB) Journal*, *7*: 163-178.

Chakrabarti S., Dom B. & Indyk P. (1998)b. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM's Special Interest Group on Data on Data Management (SIGMOD) Conference.*

Chen H. & Dumais S. (2000). Bringing order to the web: Automatically categorizing search results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 145-152).

Christianini N. & Shawe-Taylor J. (2000). *An introduction to support vector machines*. Cambridge University Press

Duda R.O. & Hart P.E. (1973). *Pattern classification and sense analysis*. Wiley & sons.

Furnkranz J. (1999). Exploring structural information for text classification on the WWW. In *Intelligent Data Analysis* (pp. 487-498).

Glover E., Tsioutsiouliklis K., Lawrence S., Pennock M. & Flake G. (2002). Using web structure for classifying and describing Web pages. In *Proceedings of the 11th International World Wide Web (WWW) Conference*.

Huang C.C., Chuang S.L. & Chien L.K. (2004). LiveClassifier: Creating hierarchical text classifiers through web corpora. In *Proceedings of the 13th International World Wide Web (WWW) Conference*, (pp. 184-192).

Kaufman L. & Rousseeuw P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & sons.

Koller D. & Sahami M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning (ICML):* (pp.170-178).

Mladenic D. (1998). Turning Yahoo into an automatic web page classifier. In *Proceedings of the 13th European Conference on Artificial Intelligence*, (pp. 473-474).

Nigam K., McCallum A.K., Thrun S. & Mitchell T.M. (2000). Text classification from labeled and unlabeled documents using EM. In *Machine Learning*, 39(2-3): 103-134

Ntoulas A., Cho J. & Olston Ch. (2004). What's new on the web? The evolution of the web from a search engine perspective. In *Proceedings of the 13th InternationalWorld Wide Web (WWW) Conference*, (pp. 1-12).

Olston Ch. & Chi E. (2003). ScentTrails: Intergrading browsing and searching. In *ACM Transactions on Computer-Human Interaction* 10, 3: 1-21.

Pazzani M. & Billsus D. (1997). Learning and revising user profiles: The identification of interesting Web sites. In *Machine Learning Journal*, 23: 313-331

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.

Ruiz M.E. & Srinivasan P. (1999). Hierarchical neural networks for text categorization. In *Proceedings of the ACM's Special Interest Group in Information Retrieval (SIGIR) Conference* (pp. 281-282).

Song Y.I., Han K.S. & Rim H.C. (2004). A term weighting method based on lexical chain for automatic summarization. In *Proceedings of the 5th Conference on Intelligent Text Processing and Computational Linguistics (CICLing*), (pp.636-639).

Stamou S., Krikos V., Kokosis P. & Christodoulakis D. (2005). Web directory construction using lexical chains. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems* (*NLD*B*).*

ENDNOTES

[1]This can be checked at: http://rdf.dmoz.org/rdf/catmv.log.u8.gz

[2]The word "ontology" and the phrase "subject hierarchy" are being used interchangeably in the chapter

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

[4]Candidate terms are nouns, verbs, adjectives or adverbs

[5]For example, net_{s1} may refer to a fishing net while net_{s2} may refer to a computer network