# Using a Web-Based Categorization Approach to Generate Thematic Metadata from Texts

CHIEN-CHUNG HUANG, SHUI-LUNG CHUANG, and LEE-FENG CHIEN
Academia Sinica, Taiwan

Conventional tools for automatic metadata creation mostly extract named entities or text segments from texts and annotate them with information about persons, locations, dates, and so on. However, this kind of entity type information is often insufficient for machines to understand the facts contained in the texts, thus precluding the possibility of implementing more advanced, intelligent applications, such as concept-based search. In this work, we try to create more refined thematic metadata inherent in texts. Based on Web resource mining, our approach acquires training corpora necessary to describe both the thematic categories and the metadata extracted from the texts. The approach then finds the corresponding relationships among them by means of categorization and thus generates thematic metadata for the textual data. Experimental results confirm the potential and wide adaptability of our approach.

Categories and Subject Descriptors: H.3.m [**Information Storage and Retrieval**]: General

General Terms: Experimentation, Measurement, Algorithms

Additional Key Words and Phrases: Categorization, metadata, Web mining

## 1. INTRODUCTION

Metadata is a useful medium for preserving and transmitting information about data, and it has been used extensively for various purposes [Gilliland-Swetland 2000]. Due to the convenience and efficiency it bestows, its usage can be traced back to a long time ago. Nowadays, computer scientists devise various kinds of mechanisms, such as XML or meta tags in HTML, to store and to utilize metatdata. Given all its usefulness, the fundamental problems of metadata still exist: (1) how to generate it efficiently, and (2) how to utilize it effectively. In this work, we will focus on the first problem.

The task of generating metadata presents two challenges: (1) how to enhance the correctness and expressiveness of metadata; (2) how to lower the cost incurred by generating it. Unfortunately, the two goals are often hard to

Authors' address: Chien-Chung Huang, Shui-Lung Chuang, and Lee-Feng Chien, Academia Sinica, Taiwan.

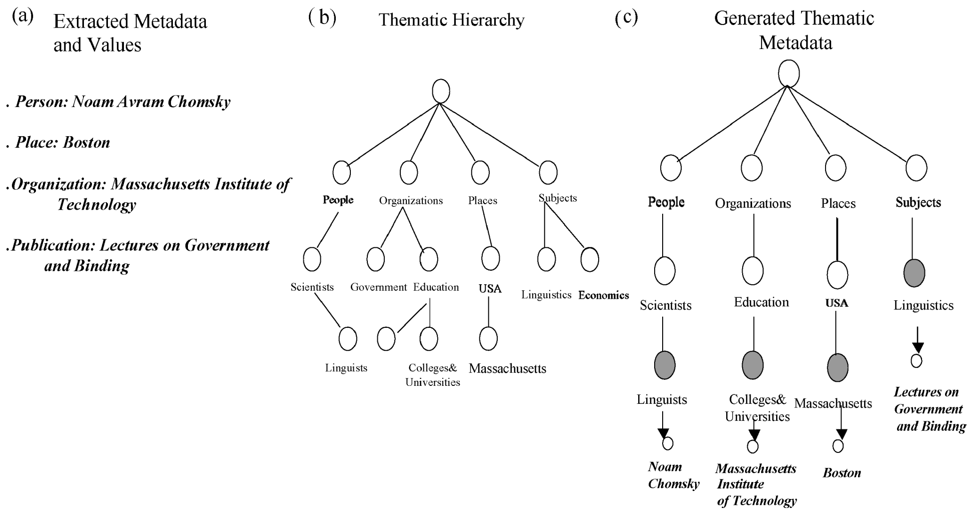


Fig. 1. Primitive metadata and values are shown in (a). In this work, we intend to categorize the value of primitive metadata (text segments) onto the thematic hierarchy shown in (b) and to create the thematic metadata indicated by the shaded circles in (c).

achieve simultaneously. If we want to create precise and semantically meaningful metadata for more advanced applications such as concept-based search or resource discovery, we unavoidably need the aid of professional librarians or indexers. However, considering the huge amount of data that already exist, manually creating metadata is obviously a too time- and labor-consuming task.

An alternative solution for metadata generation is to use some automatic mechanisms, suppose that the correctness and expressiveness of the metadata can be maintained at a certain level. In fact, recent advances in text process technologies, such as information extraction [Shlomo et al. 1998; Gao et al. 2003; Uchimoto et al. 2000; McCallum et al. 2000] and metadata annotation [Staab et al. 2001; Erdmann et al. 2000], have partly achieved this goal. For textual data, the metadata extracted and annotated using current techniques can provide the information about people, places, events, dates, and so on. However, the metadata created using these technologies, though attaining high accuracy, is still primitive and as a result is not adequate to serve as a basis for more advanced applications. Therefore, how to create more refined metadata with limited human intervention is a problem that deserves investigation.

In this work, we present a Web-based approach to creating thematic metadata automatically. Thematic metadata possesses more semantic specificity and can be taken as more refined metadata. For the purpose of illustration, we use the example presented in Figure 1 to help explain our idea. Current technologies can extract and annotate metadata as shown in Figure 1(a) from a set of documents.[1] Our goal is to discover the thematic meaning of the

[1]The metadata that we refer to here is not necessarily extracted from documents through the above-mentioned techniques. Actually, it also can be data in XML or textual data in a database. Our approach applies to these cases too.

metadata. Suppose we are given a thematic hierarchy as shown in (b), we can use a categorization technique to find out that "Boston" is not only a place, but also is located in "Massachusetts"; an analogous example is that we hope to find out "Noam Chomsky" is more than just a person, he is also a linguist.

To facilitate further discussions, we first establish some conventions when referring to certain technical terms and also briefly explain their meaning. By *primitive metadata*, we mean the data representing some broad-concept facts, such as places, dates, and people; the value of primitive metadata is *text segment*. We assume that primitive metadata is pre-given and hope to find out the *thematic metadata* inherent in them through categorization. We also assume that the *thematic hierarchy* with which the categorization is effected is pre-given. The thematic hierarchies can be those labeled by professional indexers, such as Yahoo!'s directory, or can be those temporarily constructed for special needs. A thematic hierarchy is composed of a number of *thematic categories*, each of which represents a clearly-defined concept of certain entities. Also, these thematic categories adhere to the subsumption relationship as usually defined in a hierarchical structure.

To create thematic metadata, we categorize text segments onto thematic categories. Therefore, our tasks are just like those involved in the conventional categorization (classification) problem: (1) prepare training sets to describe the thematic categories; (2) choose a proper representation model to describe the text segments [Manning and Schütze 1999]. Concerning the first task, manually labeling the training corpus is obviously not a good choice considering its costliness. As for the second task, one may find it straightforward at first sight, but in truth it is not so. One cannot simply adapt conventional document classification techniques to solve the problem. The key difference is that a document, containing thousands of (or even more) words, is usually able to provide sufficient features by itself, while a text segment, containing only a few words, is not so. Obviously, one has to resort to some other means if intending to represent text segments more richly.

Fortunately, there already exist ideal tools to accomplish both tasks: Web technologies [Kosala and Blockeel 2000]. The Web affords us with an inexhaustible source to draw the necessary corpus to characterize both the thematic categories and the text segments. What is lacking is just a well-integrated approach to fully exploiting the Web.

We here sketch the idea behind our approach and will give it a more formal treatment in Section 2. Our approach is essentially built on the ideas of following two techniques:

—A Web-based feature extraction method that exploits search-result snippets as the feature source to characterize text segments in diverse subject domains.
—An unsupervised training method that utilizes the categories names and structural information of the thematic hierarchy to automatically acquire training sets from search-result snippets for modeling the thematic categories.

Using Web resources to reinforce the insufficiency of corpora is more than finding an ad hoc substitute for manually labeled corpora. The nature of the Web itself qualifies it as a potential candidate for offering effective training corpora. Quantitatively, the Web is becoming the largest data repository in the world, providing an almost inexhaustible source from which one can draw useful information. Qualitatively, the data on the Web is constantly refreshing and enlarging hence has a higher chance of offering up-to-date and balanced information. The potential for using the Web is clearly shown in some recent works that successfully exploited the Web to solve problems in knowledge discovery, data mining, and natural language processing [Feldman et al. 1997; Soderland 1997; Volk 2001; Cohen and Singer 1996].

The main point of our approach is to use Web search-result snippets as the feature source, and use the structural information inherent in a thematic hierarchy to train the classifier. Building on this idea, we develop a technique called Hier-Concept Query Formation Method (HCQF), which generates effective queries to send to search engines to acquire a suitable training corpus. We here only sketch the idea of HCQF and leave the details to Section 2. The fundamental idea is that we measure the similarity based on the closeness of the concept of the category and that of the text segment. Our task then is to find a suitable training corpus that describes a category conceptually.

Suppose we are given a thematic hierarchy. A straightforward method of training the categories within it is to send the assigned names of the categories to search engines, and to use the returned search-result pages as the training corpus. However, considering that such a collection of Web pages must contain much noise, this simplistic method apparently can be further refined. To obtain a more satisfactory corpus, our idea is that, for a specific category, we can use its own concept in conjunction with that of its ancestral categories and descent categories to formulate more precise queries and organize the corpus more effectively. Furthermore, notice that sometimes the thematic hierarchy may not provide enough structural information itself, that is, the categories we wish to train may not have enough associated categories. In this case, we go one step further and design an automatic mechanism to find "pseudo child categories" and then use these pseudo categories to enrich the training corpus.

To evaluate the performance of our approach, extensive experiments have been conducted to categorize text segments of various domains, including subject terms, people names, times and locations, and even paper titles. The promising results demonstrate the potential and wide adaptability of our approach.

In the rest of this paper, we first introduce the proposed approach in Section 2; we then present the conducted experiments and their results in Section 3; finally, in Section 4 and 5, we discuss some related works and draw the conclusion, respectively.

## 2. THE APPROACH

We will first state the problem formally and then present our approach.

Suppose we are given a set of thematic categories, $C = \{c_1, c_2, \ldots, c_n\}$, a collection of text segments $T = \{t_1, t_2, \ldots, t_m\}$, and also a mapping $\delta : T \rightarrow 2^C$

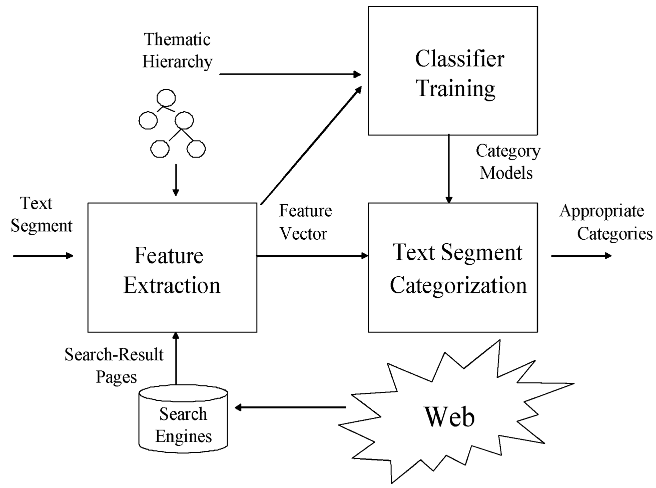


Fig. 2. An abstract diagram presenting the main components of the proposed approach.

that describes the correct category a text segment is supposed to be categorized to. Our main concern is to design a one-to-one mapping scheme $\xi : T \rightarrow 2^C$ such that the size of the correct result set $CRS = \{t_i | t_i \in T, \xi(t_i) = \delta(t_i)\}$ is the maximum.

A graphical diagram presenting our approach is shown in Figure 2. The approach is essentially composed of three computational modules: feature extraction, classifier training, and text segment categorization. The approach exploits highly ranked search-result snippets retrieved from online search engines as effective feature sources. The feature extraction module collects features for both the thematic categories and the text segment of concern, the classifier training module utilizes the feature sources to train statistical models for the thematic categories, and the text segment categorization module decides appropriate categories for the text segment. A more detailed discussion of each module is presented in the following subsections.

## 2.1 Feature Extraction and Representation

To decide the similarity between a text segment and a target thematic category, we need a representation model to describe their characteristics. As previously mentioned, a text segment cannot offer sufficient feature terms by itself, therefore precluding the possibility of computing the distance between itself and a target thematic category directly. To overcome this problem, we send the text segment as a query to search engines and use the returned pages as its feature source. Note that we use only the snippets as the source, instead of the whole page, in order to save a large number of page accesses.

We adopt the vector-space model to describe the features of both text segments and thematic categories. More precisely, as will be presented in Section 2.2, HCQF outputs a set of category objects for each separate thematic category, and both these category objects and text segments are converted into vectors, with which we can estimate the similarity between them.

Suppose that, for each query $q$ (which in fact is a text segment or some boolean expressions of category names), we collect up to $N_{\max}$ search-result snippets, denoted as $SRS_q$. Each query can then be converted into a bag of feature terms by applying normal text processing techniques, for example, removing stop words and stemming, to the contents of $SRS_q$. Let $T$ be the feature term vocabulary, and let $t_i$ be the $i$th term in $T$. With simple processing, a query $q$ can be represented as a term vector $v_q$ in a $|T|$-dimensional space, where $v_{q,i}$ is the weight $t_i$ in $v_q$. The term weights in this work are determined according to one of the conventional *tf-idf* term weighting schemes [Salton and Buckley 1988], in which each term weight $v_{q,i}$ is defined as

$$v_{q,i} = (1 + \log_2 f_{q,i})\ \log_2(n/n_i)$$

where $f_{q,i}$ is the frequency of $t_i$ occurring in $v_q$'s corresponding feature term bag, $n$ is the total number of category objects, and $n_i$ is the number of category objects that contain $t_i$ in their corresponding bags of feature terms. The similarity between a text segment and a category object is computed as the cosine of the angle between the corresponding vectors, that is,

$$\mathrm{sim}(v_a, v_b) = \cos(v_a, v_b).$$

### 2.2 Classifier Training: HCQF

For the sake of clarity, we will introduce HCQF in a step-by-step manner. A thematic hierarchy $TH = (C, R)$ is composed of two parts: thematic categories $C = \{c_1, c_2, \ldots, c_n\}$ and relations $R = \{r_1, r_2, \ldots, r_m\}$, the latter relating the former hierarchically so that super categories conceptually subsume subcategories.[2]

A thematic category, whose name is a specially assigned keyword, essentially represents an abstract concept, and the concept is usually embodied by a pre-arranged training set that describes its characteristics. For each $c_i$, if disregarding its relative position in *TH*, we can send the name of $c_i$ to search engines and use the returned snippets as its training set. We refer to a concept described by such a training set as a *general concept* $G(c_i)$ of $c_i$; however, in our case, we think that this kind of general concept is not the concept that $c_i$ is really meant to represent. The reason is that a general concept does not fully reflect the structural information inherent in the thematic hierarchy *TH*.

To address this problem, therefore, we define a *specific concept* $S(c_i)$ that we think is really the concept that $c_i$ represents in the context of a hierarchy. Our idea can be best illustrated by an example: suppose $Y$ department is a subcategory of $X$ university in some thematic hierarchy *TH*; $G(Y)$ expresses $Y$ but fails to indicate that $Y$ department is a child category of $X$ university. Instead, the concept that $Y$ really represents in *TH* not only concerns $Y$ but also contains the fact that $Y$ is a child of $X$. To put it another way, suppose we wish to train the category "CS department," which is a subcategory of "Stanford"; most of the snippets obtained using the query "CS department" necessarily contain information about "CS department," however, many of them may contain information about "CS department" of some universities other than "Stanford."

[2] In practice, we are often given a graph rather than a hierarchy. If such is the case, we "decompose" the graph into a hierarchical tree structure. In particular, if a node has multiple parents, we treat this node as if it has multiple copies and each copy has only a parent category.

Such a training set apparently is not a precise description of the category we wish to train.

Going back to the first example, in our research, $Y$'s specific concept $S(Y)$ should be the result of $Y$'s general concept $G(Y)$ constrained by its parent $X$. Following this line of reasoning, not only $X$, but also all of $Y$'s ancestors should exert some influence on $Y$'s general concept. Naturally, one is led to think of the converse. Descendant categories should also exert a reciprocal influence on ancestor categories; that is, descendant categories should enrich the concepts of ancestor categories so as to give a more precise description of them. Suppose $X$ university has three departments, $Y$, $Y'$, and $Y''$; the concept that $X$ represents not only concerns $X$ itself but also includes the fact it is the parent of $Y$, $Y'$, and $Y''$. As previously noted, all descendent categories should enrich the concept of $X$. To summarize, for each $c_i$, the content of a specific concept is determined by three factors: its ancestors, its decedents, and its own general concept.

We will now formally define a specific concept. Given some category $c_\alpha$, whose ancestors are $A_{c_\alpha}$ and whose descendants are $D_{c_\alpha}$, its specific concept, $S(c_\alpha)$, is the union of two separate parts: its specific ancestral concept, $S_A(c_\alpha)$ and its *specific descendant concept*, $S_D(c_\alpha)$, the former being its general concept constrained by its ancestors, and the latter being the unification of the specific ancestral concepts of all its descendants:[3]

$$
\begin{aligned}
S_A(c_\alpha) &= G(c_\alpha) \cap \{G(a_i) | a_i \in A_{c_\alpha}\} \\
S_D(c_\alpha) &= \cup \{S_A(d_j) | d_j \in D_{c_\alpha}\} \\
S(c_\alpha) &= S_A(c_\alpha) \cup S_D(c_\alpha).
\end{aligned}
$$

The task of preparing the training set to express $S_A(c_\alpha)$ seems difficult, but fortunately, real-world search engines relieve us of much of the trouble. One may send the query as $c_\alpha$ appended with the name of its ancestors $A_{c_\alpha}$ to search engines and use the returned snippets as the required training set for $S_A(c_\alpha)$. Conversely, when preparing the training set to express $S_D(c_\alpha)$, one can simply add up all the training sets for the specific ancestral concept of $D_{c_\alpha}$. In more practical terms, the total training set of categories $c_\alpha$ is then composed of the training set ($N_{\max}$ snippets) for $S_A(c_\alpha)$ and the training sets ($N_{\max}|D_{c_\alpha}|$) for $S_D(c_\alpha)$. Note that each $N_{\max}$ snippets can then be converted into a category object according to the vector space model discussed in Section 3.1. Therefore, $c_\alpha$ is then presented as a set of category objects, one of which is from $S_A(c_\alpha)$ while the others are from $S_D(c_\alpha)$.

The strength of HCQF lies exactly in this kind of rich training set. For $c_\alpha$, its concept is not only specified by its ancestors and itself, but also by all its descendants. Suppose we only consider $S_A(c_\alpha)$ and drop $S_D(c_\alpha)$; we have merely $N_{\max}$ snippets to train $c_i$. On the other hand, if we take $S_D(c_\alpha)$ into consideration, we may have a dramatic $(1 + |D_{c_\alpha}|)N_{\max}$ snippets to train $c_\alpha$. (Note that since we convert the training sets into category objects, the above expression can be better restated as a comparison between 1 and $1 + |D_{c_\alpha}|$ objects.) Moreover, the

[3]We use the set operations to express our idea, and their meaning will be made clear in the following discussion. Strictly speaking, the notation we use here is not mathematically rigorous. The "concept" actually is not composed of distinct entities as a mathematical set is.

additional $|D_{c_\alpha}|N_{\max}$ snippets (or $|D_{c_\alpha}|$ objects) do not contain as much noise as one might expect for they are already constrained by $c_\alpha$ along with its ancestors.

It can easily be seen that one may train all the categories in $TH$ by traversing it in the manner of BFS (breadth-first-search) twice. For each $c_i$, the first pass collects the training set for $S_A(c_i)$, while the second pass adds up the training sets for $S_D(c_i)$ and $S(c_i)$. Having collected all the necessary training sets, by using the collection of the terms in all the training sets as the total feature vocabulary $T$, we can convert these training sets into a set of category objects.

We now are left with two more questions. First, the leaf category $c_{\text{leaf}}$, unlike internal categories, cannot be strengthened by its descendent categories. In other words, $S_d(c_{\text{leaf}}) = \emptyset$, and HCQF seems to have a weaker ability to describe them. Second, suppose we are given only a nonhierarchical tree, that is, a flat structure. Can HCQF be generalized so that it can be applied to the nonhierarchical tree too? The answer to both questions lies in the fact that one can always find some categories to enrich a leaf category by inserting them as "pseudo" children. Given some leaf category $c_{\text{leaf}}$, when collecting snippets to organize $S_a(c_{\text{leaf}})$, one can easily find some associated terms of $c_{\text{leaf}}$ and use some filtering mechanisms to choose proper terms as child categories of $c_{\text{leaf}}$.

In our experiments, we employed the following two techniques to create "pseudo" child categories. We chose either (1) the terms subsumed by $c_{\text{leaf}}$ [Sanderson and Croft 1999] or (2) the terms that have the highest mutual information with $c_{\text{leaf}}$.

The first technique is based on the following assumption: suppose a term $c^d_{\text{leaf}}$ is subsumed by $c_{\text{leaf}}$; the documents in which $c^d_{\text{leaf}}$ always (or almost always) appears contain $c_{\text{leaf}}$, while the documents containing $c_{\text{leaf}}$ do not necessarily contain $c^d_{\text{leaf}}$. According to Sanderson and Croft [1999], the formula should be set as follows:

$$P\left(c_{\text{leaf}}|c^d_{\text{leaf}}\right) \geq \sigma$$
$$P\left(c^d_{\text{leaf}}|c_{\text{leaf}}\right) < 1.$$

Sanderson and Croft [1999] set $\sigma$ to 0.8. However, due to the difference in experimental environments (our documents are derived from open source), we found that 0.85 would be a more suitable setting for $\sigma$.

The second technique is based on the assumption that the concept of $c_{\text{leaf}}$ can be enriched by its most relevant terms, too. We choose the terms that appear with $c_{\text{leaf}}$ more than a certain number of times. We denote the document frequency of some term $t$ as $DF(t)$, and that of the co-occurrence of $s$ and $t$ as $DF(t, s)$; then our idea can be expressed as

$$\frac{DF(t, c_{\text{leaf}})}{DF(c_{\text{leaf}})} > \epsilon.$$

In all experiments, we set $\epsilon$ as 0.45. The whole algorithmic procedure of HCQF is presented in Figure 3.

## 2.3 Text Segment Categorization

Given a new text segment $t$, we have to determine a set of categories $C_t$ that are considered as $t$'s most related categories. As discussed in Section 2.1, the

**HCQF**$(TH = (C, R))$
$C$: the set of thematic categories
$R$: relations among thematic categories
$SRS$: the collection of training sets (search result snippets) (initially ø)
$CO$: the collection of feature vectors of category objects
1: **for all** $c \in C$, according to $R$, choose $c$ by BFS **do**
2: $SRS_c \leftarrow$ send $c \cup c_{\text{ancestors}}$ to search engines
3: **if** c= leaf **then**
4: $AT_c \leftarrow$ Associated_Terms_by_Subsumption($c$, $SRS_c$)
5: **for all** $at \in AT_c$ **do**
6: $SRS_c \leftarrow SRS_c \cup$ send $(c \cup at)$ to search engines
7: **end for**
8: **end if**
9: **end for**
10: **for all** $c \in C$, according to $R$, choose $c$ by BFS **do**
11: $SRS_c \leftarrow SRS_c \cup SRS_{c_{\text{descendents}}}$
12: $SRS \leftarrow SRS \cup SRS_c$
13: **end for**
14: **for all** $srs \in SRS$ **do** {transform SRS into a feature vector according to Section 3.1}
15: $co \leftarrow$ transform $srs$
16: $CO \leftarrow CO \cup co$
17: **end for**
18: return $CO$
**Associated_Terms_by_Subsumption**($c$, $SRS_c$)
$c$: thematic category
$SRS_c$: training set of $c$
$AT$: the set of associated terms (initially ø)
19: **for all** $t \in SRS_c$ **do**
20: **if** $p(c|t) \geq 0.85$ and $p(t|c) < 1$ **then**
21: $AT \leftarrow AT \cup t$
22: **end if**
23: **end for**
24: return $AT$
**Associated_Terms_by_Co-occurrence**($c$, $SRS_c$)
$c$: thematic category
$SRS_c$: training set of $c$
$AT$: the set of associated terms (initially ø)
25: **for all** $t \in SRS_c$ **do**
26: **if** $DF(t, c)/DF(C) > \varepsilon$ **then** {$DF$ value can be obtained from search engines}
27: $AT \leftarrow AT \cup t$
28: **end if**
29: **end for**
30: return $AT$

Fig. 3. An algorithmic procedure describing HCQF. Note that line 4 can be replaced by Associate_Term_by_Co-occurence($c$, $SRS_c$).

candidate text segment $t$ is represented as a feature vector $v_t$. For this categorization task, we adopt a kNN approach.

kNN has been found to be an effective classification approach to a broad range of pattern recognition and text classification problems [Dasarathy 1991]. Using the kNN approach, a relevance score between $t$ and candidate category object $C_i$ is determined by the following formula:

$$r_{\text{kNN}}(t, C_i) = \sum_{v_j \in R_k(t) \cap C_i} \text{sim}(v_t, v_j)$$

**TextSegmentCategorization**$(t, C, CO, n)$
$t$: the unknown text segment
$C$: the set of categories
$CO$: the set of category objects returned by **HCQF**
$n$: the number of target categories

**for all** $co \in CO$ **do**
  $r(t, co) \leftarrow \mathrm{sim}(v_t, v_co)$
**end for**
$R_k(t) \leftarrow$ the $k$ category objects $co \in CO$ with highest $r(t, co)$ scores
**for all** $c \in C$ **do**
  $r_{\mathrm{kNN}}(t, c) \leftarrow 0$
**end for**
**for all** $co \in R_k(t)$ **do**
  **for all** $c$ is a category of $co$ **do** {a category object $co$ may enrich multiple ancestor categories}
    $r_{\mathrm{kNN}}(t, c) \leftarrow r_{\mathrm{kNN}}(t, c) + r(t, dv)$
  **end for**
**end for**
return the top-ranked $n$ categories in $C$ according to the decreasing order of $r_{\mathrm{kNN}}(t, c)$

Fig. 4. An algorithmic procedure describing the text segment categorization process.

where $R_k(t)$ represents $t$'s $k$ most-similar category objects, measured by a "sim" function, in the whole collection. Figure 4 shows the algorithmic procedure of this categorization process.

The categories that a text segment is to be assigned to are determined by either a predefined number of most-relevant clusters or a threshold used to pick those clusters that have scores higher than the specified threshold value. Different threshold strategies have both advantages and disadvantages [Yang 2001]. In this study, to evaluate the performance of our approach, we select the five most-relevant categories as candidates.

## 3. EXPERIMENTS

To assess the performance of our approach, we conducted a series of experiments. In Section 3.1, we examine whether HCQF can be really justified by empirical evidence. After that, we shift our focus to other possible applications.

In general, we are interested in the following questions and designate them respectively as Approaches 1–4:

(1) Suppose we use only the restricted version of HCQF, that, suppose we drop $S_D(C)$ and only use $S_A(C)$, remove the third-level categories, and do not consider generating pseudo child categories. How well does HCQF perform? This is equivalent to using only a flat structure and can be thought of as the bottom-line. (Approach 1)
(2) Suppose we take both $S_D(C)$ and $S_A(C)$ into consideration but still do not generate pseudo categories automatically. How well does HCQF perform? (Approach 2)
(3) Suppose we are not given third-level categories. How well does HCQF perform if it generates pseudo categories automatically using the subsumption technique? (Approach 3)

(4) The same situation as (3), but now HCQF generates pseudo categories using the co-occurrence technique. (Approach 4)

In all tables, the Top-$n$ inclusion rate is the rate of test text segments whose highly ranked $n$ candidate categories contain the correct category. In most cases, we used Yahoo!'s directory as the testing bed. When needs arose, we also used thematic hierarchies and testing text segments from other sources to conduct the experiments. All data not based on Yahoo!'s directory can be found in WKD [2003]. The Web search engine we used in these experiments was Google, $N_{\max}$ was set to 100 and $k$ was set to 5, unless the caption of the tables specifies otherwise.

### 3.1 Overall Performance of the Proposed Approach

We first observed the performance of HCQF under various circumstances. In designing the experiments, we shifted the focus progressively to better examine every facet of HCQF.

3.1.1 *Under the Condition of a Well-Designed Thematic Hierarchy: Yahoo!'s Computer Science Tree.* Suppose we were given a set of text segments extracted from a Computer Science archive. We wished to know what kind of thematic metadata could be created by our approach. Therefore, we designed the following experiment.

In the Yahoo! Computer Science tree (CS thematic hierarchy), there were totally 36 second-level, 177 third-level, and 278 fourth-level categories, all rooted at the category "Computer Science." The fourth-level categories, such as "Intelligent Software Agent" and "Fuzzy Logic," could be taken as primitive metadata values. We decided to use the second-level categories, for example, "Artificial Intelligence" and "Linguistics," as the target thematic categories and attempted to categorize these fourth-level category names (text segments) onto them, thereby creating the corresponding thematic metadata.

From Table I, it can be observed that Approach 2 surpassed all the other approaches. This suggested that a well-organized thematic hierarchy greatly contributed to the high performance of HCQF. Approaches 3 and 4 also produced good results though not as good as those of Approach 2. This indicated that both the subsumption technique and the co-occurrence technique could create proper pseudo categories to some degree. The worst results were obtained by Approach 1. However, it should be noted that although this approach fell far behind other approaches, its overall performance was still acceptable (considering that there were totally 36 categories). This reveals the quality of Web resources.

After examining whether the test text segments could be successfully categorized onto second-level categories, we were also curious about how HCQF performed when test text segments were to be categorized onto third level categories. Categorizing test text segments onto different levels of the thematic hierarchy is essentially creating thematic metadata of different degrees of refinement. We discuss this issue further in Section 3.1.6.

Out of 177 third-level categories, 61 of them had child categories and the 278 test text segments were to be categorized onto them. Table II lists the results. It

Table I. Categorizing Fourth-Level Category Names Onto Second-Level Categories of a CS Thematic Hierarchy from Yahoo!

| | Approach | # of Total (Distinct) Words | # of Effective (Distinct) Words | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|---|
| Manual hierarchy | Based on only second-level categories (Approach 1) | 98,435 (11,647) | 58,115 (3140) | 0.4917 | 0.6346 | 0.6943 |
| | Based on first three level categories (Approach 2) | 600,749 (39,579) | 385,839 (10,862) | 0.7409 | 0.8605 | 0.9103 |
| Augmented hierarchy | Based on pseudo categories generated by the subsumption technique plus categories at the first two levels (Approach 3) | 447,838 (30,747) | 285,709 (9204) | 0.6060 | 0.7252 | 0.8146 |
| | Based on pseudo categories generated by the co-occurrence technique plus categories at the first two levels (Approach 4) | 500,729 (31,838) | 322,766 (9768) | 0.5949 | 0.7342 | 0.7807 |

Number of target categories = 36. Number of test text segments = 278. Number of created pseudo child categories = 4. Effective words in the fourth column mean non-stop words and those whose document frequency is higher than 3.

Table II. Categorizing Fourth-Level Category Names Onto Third-Level Categories of the CS Thematic Hierarchy from Yahoo!'s Directory

| Approach | # of Total (Distinct) Words | # of Effective (Distinct) Words | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|
| Approach 1 | 194,785 (19,907) | 111,078 (5446) | 0.4493 | 0.6419 | 0.6993 |
| Approach 3 | 801,160 (48,159) | 499,135 (14,137) | 0.4932 | 0.6993 | 0.7364 |
| Approach 4 | 982,443 (53,866) | 610,836 (15,944) | 0.5034 | 0.6587 | 0.7399 |

Number of target categories = 61. Number of test text segments = 278. Number of created pseudo child categories = 4. The absence of Approach 2 is due to the fact that all fourth-level categories were now used as test text segments.

can be observed that the results were still promising (considering again the difficulty of the task—there were now 61 target categories.) The results suggested that HCQF was not restricted to be applied to a certain level of the thematic hierarchy.

3.1.2 *Categorization by Directly Using Term Occurrence in the Training Corpora.* Uncertain about the fundamental difficulty of the thematic metadata generation, we devised a reference experiment to complement the preceding CS experiment. We thought, maybe using a very straightforward strategy could also achieve good results, and if this was so, we did not have to bother about our more circuitous approach.

As before, we used Approach 1 to collect the training corpora. But we did not use vector space model to compute the similarity between a text segment and a thematic category. Rather, we simply computed how many times a text segment appeared in the corpora of each thematic category. The categories with most corresponding text segment appearances were thought to be the corresponding thematic metadata.

Table III. Top-*n* Inclusion Rate of Categorizing Text Segments by Simply Counting their Appearance in the Training Corpora

| Top-1 | Top-2 | Top-3 |
|---|---|---|
| 0.1063 | 0.2990 | 0.3189 |

The corpora was organized using Approach 1.

Table IV. Comparing the Results of HCQF given Different Size of Training Corpus

| Approach | # of Snippets | # of Total (Distinct) Words | # of Effective (Distinct) Words | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|---|
| Approach 1 | 100 | 98,435 (11,647) | 58,115 (3140) | 0.4917 | 0.6346 | 0.6943 |
| | 75 | 74,055 (9469) | 43,241 (2519) | 0.5116 | 0.7143 | 0.7741 |
| | 50 | 49,517 (7169) | 28,181 (1862) | 0.5216 | 0.6611 | 0.7375 |
| | 25 | 17,509 (3619) | 8895 (764) | 0.4950 | 0.6412 | 0.7276 |
| Approach 2 | 100 | 600,749 (39,579) | 385,839 (10,862) | 0.7409 | 0.8605 | 0.9103 |
| | 75 | 451,072 (32,450) | 286,873 (8908) | 0.7375 | 0.8671 | 0.9070 |
| | 50 | 300,742 (24,512) | 188,885 (6817) | 0.7442 | 0.8671 | 0.9003 |
| | 25 | 87,612 (11,034) | 51,888 (2863) | 0.7342 | 0.8604 | 0.8937 |

Number of target categories = 36. Number of test text segments = 278.

Table III lists the results, which, compared to Table I, was obviously not satisfactory.

3.1.3 *Differing the Number of Snippets to Train the Thematic Hierarchies.* Continuing the preceding Computer Science experiment, we were concerned about whether the number of snippets used to train the thematic categories was a decisive factor in the performance of HCQF.

Table IV lists the results using both Approach 1 and 2. When the number of snippets extracted from the search engine is below 100, the performance remained in general stable. It can be observed that the size of the training corpora is not a decisive factor in training the thematic hierarchies. This observation has grave implications on the nature of HCQF. We postpone further discussions about the size of training corpora till Section 5.

3.1.4 *Search Engines Comparison.* Again continuing the Computer Science experiment, we compared the performance of HCQF using different Web search engines. Table V lists the results of using both Google and Lycos. It can be observed that Google had the better of Lycos, though the latter still attained reasonable results. A specific "good" search engine was not absolutely necessary; however, the performance HCQF was unavoidably influenced by the choice search engine.

3.1.5 *Ad hoc Thematic Hierarchies.* In all preceding experiments HCQF had been applied to a well-defined thematic hierarchy—Yahoo!'s CS thematic hierarchy. We then examined whether HCQF could also be applied to not thoroughly well-defined thematic hierarchies, that is, those thematic hierarchies not created by professional indexers.

We asked two students respectively of EE (Electrical Engineering) and of Physics departments to create thematic hierarchies encompassing the main

Table V. Comparing the Results of HCQF Using Different Search Engines

| Search Engines | Approach | # Total (Distinct) Words | # Effective (Distinct) Words | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|---|
| Google | Approach 1 | 98,435 (11,647) | 58,115 (3140) | 0.4917 | 0.6346 | 0.6943 |
| Lycos | Approach 1 | 102,045 (2588) | 70,850 (2150) | 0.2990 | 0.3887 | 0.4684 |
| | Approach 2 | 692,958 (7733) | 452,574 (7349) | 0.6445 | 0.7807 | 0.8306 |
| | Approach 3 | 490,564 (6406) | 352,895 (5984) | 0.3588 | 0.4751 | 0.5548 |
| | Approach 4 | 502,123 (7032) | 340,021 (5534) | 0.4120 | 0.5116 | 0.5614 |

Number of target categories = 36. Number of test text segments = 278. Note that the "Google" row is identical with the results in Table I.

Table VI. Applying HCQF to Ad-hoc EE and Physics Thematic Hierarchies

| Areas | # of Categories | # of Test Segments | Approach | # Total (Distinct) Words | # Effective (Distinct) Words | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|---|---|---|
| EE | 9 | 45 | Approach 1 | 23,072 (4175) | 12,958 (912) | 0.5714 | 0.7619 | 0.8095 |
| | | | Approach 3 | 88,889 (9706) | 55,617 (2826) | 0.6904 | 0.8095 | 0.8571 |
| | | | Approach 4 | 121,529 (11,131) | 78,363 (3466) | 0.6190 | 0.7619 | 0.8571 |
| Physics | 14 | 53 | Approach 1 | 35,825 (5490) | 20,817 (1250) | 0.3457 | 0.5062 | 0.7284 |
| | | | Approach 3 | 159,970 (14,074) | 103,387 (3942) | 0.3580 | 0.5185 | 0.7531 |
| | | | Approach 4 | 166,480 (12,538) | 107,721 (3993) | 0.3457 | 0.5309 | 0.7531 |

Number of created pseudo child categories = 4.

fields of their disciplines. We also asked them to choose some technical terms as the test text segments. For example, in the "Physics" case, the metadata "Active Galactic Nuclei" was supposed to be categorized onto the thematic category "Astrophysics"; an analogous example for "EE" case was that "VLSI chip design" was supposed to be categorized onto "Integrated Circuits and Systems."

Table VI lists the results. As the two students only offered a flat structure and as a result Approach 2 was missing in this experiment. The EE experiment got good results; however, the Physics case suffered to some extent. The reason seemed to be that in the Physics thematic hierarchy, the concept of many target categories overlapped, for example, "Quantum Computing" and "Quantum Electronics"; hence it was difficult to train them correctly. This corroborates with the previous conclusion that a well-defined hierarchy is instrumental in ensuring that HCQF performs well.

3.1.6 *Granularity and Diversity.* Having observed how HCQF performed some specific areas, Computer Science, EE, and Physics, we then tried to examine whether HCQF could be applied to a thematic hierarchy of more diverse domains and of deeper depth. We extracted parts of Yahoo!'s directory about "Science" and "Social Science" of five-level deep. There were totally 84 test text segments in Level 5 and we tried to categorize them onto Level 2, 3, and 4, respectively.

Table VII lists the details about the thematic categories from the Level 2 to Level 4.

As discussed in Section 3.1.1, categorizing text segments onto different levels of the thematic hierarchy has a consequential implication: it means HCQF can create thematic information of different degrees of refinement. In particular, the depth of a category in a thematic hierarchy suggests its own topicality and speciality. If a test text segment can be successfully categorized onto categories of different levels, it means much more information can be thus created.

Table VII. The Information of Science and Social Science Hierarchies Extracted from Yahoo!'s Directory

| Level 2 | Level 3 | Level 4 |
|---|---|---|
| Science | Mathematics | Geometry, Number Theory |
| | Chemistry | Chemist, Chemical and Biological Weapons |
| | Astronomy | Solar System, Cosmology |
| Social Science | History | Historiology, Genealogy |
| | Sociology | Social Class and Stratification, Urban Studies |
| | Linguistics and Human Languages | Translation and Interpretation, Word and Wordplay |

Table VIII. The Top 1 Inclusion Rate of Categorizing Test Text Segments in Level 5 Onto Different Levels of Thematic Categories

| Approach | Level 2 | Level 3 | Level 4 |
|---|---|---|---|
| Approach 1 | 0.4609 | 0.5714 | 0.8810 |
| Approach 2 | 0.8333 | 0.9634 | N/A |
| Approach 3 | 0.6190 | 0.6428 | 0.9166 |
| Approach 4 | 0.5827 | 0.6310 | 0.9048 |

Number of pseudo child categories = 4. Number of test text segments = 87.

Table VIII shows the results. Apparently, the diversity of the content of the thematic hierarchy did not have a strong influence on the performance, implying the HCQF can be applied to very general domain(s).

It also can be observed that categorizing test segments onto different levels of the thematic hierarchy usually got good results. One might have the impression that the higher the levels, the fewer the categories and it might appear easier to categorize test text segments onto them. However, this factor was canceled by another fact: the concept of higher level categories was harder to be trained correctly due to their generality and abstraction. We shall encounter the similar happening in the Time experiment in Section 3.2.

3.1.7 *Chinese Experiment*. We were also concerned about whether HCQF could get reasonable results when the experiments were done in different languages. We conducted another experiment in traditional Chinese. Unlike previous experiments that we used single word as feature terms, in this experiment we treated single bi/tri gram as features. Again, we used Yahoo!'s local directory of Taiwan as the testing bed. We extracted totally 30 categories from "Natural Science" (Science/Natural_Science) and of "Social Science" (Science/Natural_Science) and made them a three-level deep "Science" thematic hierarchy. We then used the third-level categories as the test text segments and categories them onto the second level. We could infer from the results in Table IX that HCQF was not language dependent.

### 3.2 Thematic Metadata Generation for Named Entities

Having evaluated the overall performance of the proposed approach, we then wanted to know whether it could be adapted to other purposes. In the fields of Natural Language Processing and of Information Extraction, the information about places, people, and dates are often the main concerns. We then decided to apply the proposed approach to them.

Table IX. Categorizing Test Text Segments Onto Second-level Science Thematic Hierarchy in Chinese Environment

| Approach | # Total (Distinct) Words | # Effective (Distinct) Words | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|
| Approach 1 | 148,352 (59,221) | 88,955 (9154) | 0.6143 | 0.8143 | 0.8929 |
| Approach 3 | 1,105,579 (283,653) | 835,392 (61,415) | 0.7 | 0.8428 | 0.9214 |
| Approach 4 | 905,475 (240,546) | 676,535 (571,07) | 0.6643 | 0.85 | 0.9214 |

Number of created pseudo child categories = 5. Number of target categories = 30. Number of test text segments = 140.

Table X. The Information of the Three Thematic Hierarchies Extracted from Yahoo!

| Thematic Hierarchy | # 2nd-Level Categories (Target Categories) | # 3rd-Level Categories | # 4th-Level Categories (Test Text Segments) |
|---|---|---|---|
| People (People/Scientist) | 9 | 156 | 130 |
| Place (Region/Europe) | 44 | 100 | 116 |
| Time(History/By_Time_Period) | 8 | 274 | 93 |

Table XI. Top 1–3 Inclusion Rates for Categorizing Yahoo!'s People, Place, and Time Category Names

| Thematic Hierarchy | Approach | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|
| People | Approach 1 | 0.6154 | 0.8154 | 0.9 |
| | Approach 2 | 0.7077 | 0.8308 | 0.9231 |
| | Approach 3 | 0.7846 | 0.8992 | 0.9077 |
| | Approach 4 | 0.7462 | 0.8692 | 0.9230 |
| Place | Approach 1 | 0.9482 | 0.9569 | 0.9655 |
| | Approach 2 | 0.9482 | 0.9655 | 0.9655 |
| | Approach 3 | 0.9482 | 0.9655 | 0.9655 |
| | Approach 4 | 0.9482 | 0.9655 | 0.9655 |
| Time | Approach 1 | 0.1612 | 0.3225 | 0.4301 |
| | Approach 2 | 0.3854 | 0.5521 | 0.6354 |
| | Approach 3 | 0.2872 | 0.4358 | 0.5038 |
| | Approach 4 | 0.2433 | 0.4020 | 0.5304 |

Test text segments were drawn from the faculties of U.S. universities, Google's directory (People), Altavista's directory (Place) and Yahoo!'s directory (Time). Number of created pseudo child categories = 4.

We conducted three more experiments, based on the thematic hierarchies extracted from Yahoo!, including "People" (People/Scientist), "Place" (Region/Europe), and "Time" (History-time Period); Table XI lists the categorization results and Table X the detailed information about these thematic hierarchies and the test text segments used.

Unlike the previous experiments, for the "People" and "Place" experiments, we used open testing data. For the "People" experiment, we collected the names of the faculty of linguistics department of University of Pennsylvania and of physics department of Stanford University, and also of the mathematicians listed in Google's directory (Science/Math/Numerical_Analysis/People) as the test text segments. HCQF still got high accuracy in open test. It should be pointed out in this experiment the scientists used as test text segments in general did not have very rich collections of relevant pages on the Web and it was relatively harder to collect enough Web pages to describe them. This may partly explain the anomalous results that Approaches 3 and 4 got the better of

Table XII. The Information of the Paper Data Set

| Conference | # Papers | Assigned Category |
|---|---|---|
| AAAI'02 | 29 | CS: Artificial Intelligence |
| ACL'02 | 65 | CS: Linguistics |
| ICML'02 | 87 | CS: Artificial Intelligence |
| JCDL'02 | 69 | CS: Lib. & Info. Sci. |
| SIGCOMM'02 | 25 | CS: Networks |
| SIGGRAPH'02 | 67 | CS: Graphics |

Approach 2. The chief reason that Approach 2 did not get best results as before was, we conjecture, that the Web pages of those classical scientists (training corpora in sub-level categories) and the Web pages of more recent researchers (test text segments) were quite different in content. The relevant Web pages about the former were of more biographical nature while those describing the latter more emphasized on their publication and research work.

For the "Place" experiment, we randomly collected 116 places names from AltaVista's directory (Regional/Europe) and categorized them. It can be observed that the accuracy remained as high as before, suggesting that the high accuracy of HCQF in "Place" experiment could still be attained in open tests.

For the "Time" experiment, we picked up 93 names, which could be considered as text segments, from the Yahoo!'s bottom-level and assigned them to the second-level categories so as to create their corresponding thematic metadata. It can be observed in the "Time" case we did not get similarly good results. The reason seemed to be that the concept of a time period, such as "Renaissance" or "Middle Ages," was too broad, therefore, too much noise was unavoidably contained in the returned snippets, thus impairing the quality of the training corpora and lowering the precision of categorization.

### 3.3 Thematic Metadata Generation for Longer Text Segments

We had so far considered only short text segments, but the text segments of interest may be longer. We were therefore curious about whether our approach could be applied to them too. To this end, we designed another experiment.

Continuing the scenario described in Section 3.1.1, we supposed that from a Computer Science archive we extracted some paper titles (long text segments). We wished to create thematic metadata from them.

We collected a data set consisting of the academic paper titles from six computer science conferences held in 2002 and tried to categorize them onto the 36 second-level CS categories as described in Section 3.1.1. Each conference was assigned to the Yahoo! category to which the conference was considered to belong; for example, AAAI'02 was assigned to "Computer Science/Artificial Intelligence," and all the papers from that conference were unconditionally assigned to that category. Table XII lists the relevant information of this paper data set. Notice that this might not be an absolutely correct categorization strategy as some papers in a conference may be more related to other domains than the ones we assigned them. However, to simplify our experiment, we made this straightforward assumption. Table XIII lists some "wrongly" categorized examples and it can be observed that the miss-categorized papers were often more related to the result thematic categories than those we assigned them.

Table XIII. Selected Examples of Miss-Categorized Paper Titles

| Paper Title | Conference | Target Cat. | Top-1 | Top-2 | Top-3 |
|---|---|---|---|---|---|
| A New Algorithm for Optimal Bin Packing | AAAI | AI | ALG | AI | MOD |
| (Im)possibility of Safe Exchange Mechanism Design | AAAI | AI | NET | SC | LG |
| Performance Issues and Error Analysis in an Open-Domain Question Answering System | ACL | LG | AI | LG | ALG |
| Active Learning for Statistical Natural Language Parsing | ACL | LG | AI | LG | NN |
| Improving Machine Learning Approaches to Coreference Resolution | ACL | LG | AI | LG | ALG |
| A Language Modelling Approach to Relevance Profiling for Document Browsing | JCDL | LIS | AI | UI | LG |
| Structuring Keyword-based Queries for Web Databases | JCDL | LIS | AI | LIS | DB |
| A Multilingual, Multimodal Digital Video Library System | JCDL | LIS | LG | UI | LIS |
| SOS: Secure Overlay Services | SIGCOMM | NET | SC | NET | MC |

AI, Artificial Intelligence; ALG, Algorithms; ARC, Architecture; COLT, Computational Learning Theory; DB, Databases; MOD, Modeling; NET, Networks; NN, Neural Network; DNA, DNA-Based Computing; FM, Formal Methods; LG, Linguistics; LIS, Library and Information Science; MC, Mobile Computing; OS, Operating Systems; SC, Security; UI, User Interface.

Table XIV. Top 1–5 Inclusion Rates for Categorizing Paper Titles

| Approach | Top-1 | Top-2 | Top-3 |
|---|---|---|---|
| Approach 1 | 0.1667 | 0.2398 | 0.3129 |
| Approach 2 | 0.5906 | 0.7544 | 0.8012 |
| Approach 3 | 0.2573 | 0.3830 | 0.4503 |
| Approach 4 | 0.2310 | 0.3743 | 0.4591 |

Number of target categories = 36. Number of (long) test text segments = 342. Number of created pseudo child categories = 4.

Tables XIV and XV list the experimental results. The level of accuracy achieved showed that our approach worked well with longer text segments. Moreover, the results suggested that our approach was versatile: besides generating thematic metadata, it can also be used to organize “things,” as long as each “thing” has a sufficient corpus on the Web to describe it.

Based on the results of the above experiments, it can be observed that the goal we set in Figure 1 was achievable. By categorizing the persons, places and paper titles onto corresponding thematic categories, thematic metadata from the text concerning “Noam Chomsky” was successfully generated.

### 3.4 Possible Applications

Having described our experiments on creating thematic metadata, in this section, we will discuss some of its possible applications. Although it is not easy to quantify their usability and effectiveness, we will sketch our ideas in the hope of giving some insight into the potential practical uses of our approach.

Table XV. Top 1–5 Inclusion Rates for Each Conference (Approach 2)

| Conference | Top-1 | Top-2 | Top-3 |
|---|---|---|---|
| AAAI'02 | 0.4138 | 0.6552 | 0.6897 |
| ACL'02 | 0.5231 | 0.7538 | 0.8308 |
| ICML'02 | 0.4483 | 0.6207 | 0.6897 |
| JCDL'02 | 0.6956 | 0.8261 | 0.8696 |
| SIGCOMM'02 | 0.92 | 1.0 | 1.0 |
| SIGGRAPH'02 | 0.6866 | 0.8060 | 0.8209 |

3.4.1 *Concept-Based Search.* One of the merits of thematic metadata is that it allows more advanced search options. We designed a computer science database of experimental nature. The database consists of the 342 data records drawn from the Paper Title experiment we have discussed in Section 3.3. Each record contains a paper title as well as its first author's name and affiliation. Besides creating the database, we also merged the People, Place, and CS thematic hierarchies used in our previous experiments and created a task-oriented thematic hierarchy. For all records, we then categorized the data listed in each column, that is, the paper title, the first author's name, onto the hierarchy we had constructed. More precisely, we used the authors' names and affiliations as text segments and categorized them onto the third-level classes, thereby creating meta-information for each of the second-level classes. For example, for the paper title "Watch their Moves: Applying Probabilistic Multiple Object Tracking to Autonomous Robot Soccer" (AAAI '02), we categorized its first author's name onto the category "Germany"; thus, the database derived new meta-information "place ='Germany'" for this record. Similarly, the paper title was categorized onto the category "Robotics,'" and the database received more meta-information "Expertise= 'Robotics'" and so on.

This added meta-information could allow users to give more refined, more advanced SQL queries, such as "Search for all the people in Germany with Robotics background." The categorization accuracy of the generated meta-information was close to that of the previous experiments, confirming our belief that the proposed approach can be useful for concept-based search.

3.4.2 *Information Summarization.* Thematic metadata can also be thought of as another form of summarization, presenting a deeper meaning of important facts extracted from the texts. Given an article of unknown nature, we can extract some facts from it and categorize them. Thus, we may grasp their intended meaning and thereby have a clearer understanding of the whole article. We will use an example to illustrate our point. After randomly selecting several biographical pages about scientists (these pages can be fetched from Yahoo!), we then extracted some facts (keywords) from them manually and attached them to the thematic hierarchy composed of the subhierarchies concerning People, Place, and Time that we had organized previously. Note that in this example a fact could be categorized onto multiple categories. After stemming the branches without attached facts, the thematic hierarchy with the categorized facts could then reflect the topic structure of the information contained in the facts. As shown in Figure 5, this kind of thematic hierarchy offers a new perspective to understand an article.

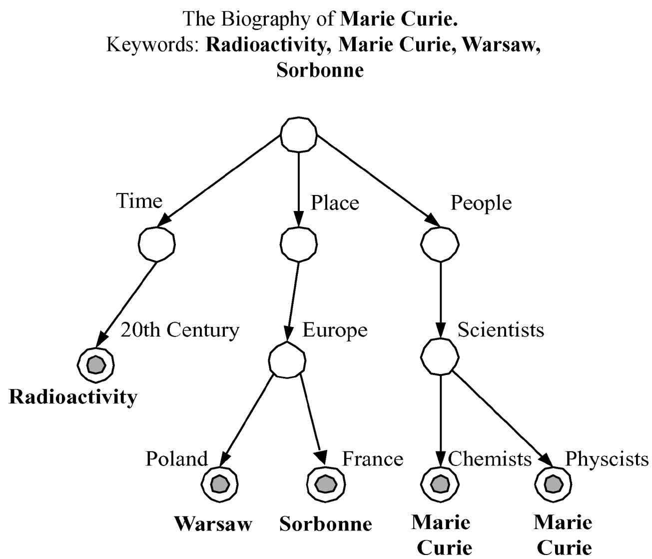


Fig. 5. A thematic hierarchy with categorized facts.

Conventionally, when confronting a new document, one will try to classify it and judge its contents based on the nature of the assigned class. However, we tackle this problem from a different angle: we categorize its important facts. Although it is hard to compare the effect of document classification with that of (extracted) text categorization, we would like to stress that the latter has the advantage of being more precise, giving a more concrete concept; on the other hand, a document's contents may be ambiguous, and sometimes it is hard to classify it correctly.

## 4. RELATED WORK

*Metadata Generation*: Conventionally, metadata generation mostly relied on manual labor, though recently there had been some attempts to make the process semi-automatic [Erdmann et al. 2000; Staab et al. 2001]. To the best of our knowledge, how to generate thematic metadata automatically has never been considered in the literature. The work most closely related to ours is that of Jenkins et al. [1999]. Superficially, their idea is quite similar to ours in that they also try to create metadata through classification. They first classify HTML documents through an hierarchical filter and decide the most closely related classes of the documents. Their metadata in essence is the by-product of the process of classification. For example, when comparing the distance between a document and classes, they extract the author, the keywords, and the word count of this document, and these are treated as metadata to be presented in RDF [1997]. It can be observed that the classification (categorization) process involved in our work is quite different from theirs. In addition, instead of classifying the whole HTML documents, we categorized extracted primitive metadata.

*Text Categorization*: Text categorization techniques are often utilized to analyze relationships among documents [Cohen and Singer 1996], but as we mentioned previously, there is a great difference between document categorization and text segment categorization. Documents normally contain more information than text segments do. The similarity between a document and a target class can be estimated based on the difference in the distribution of the words contained in document itself and the training set of the class; on the other hand, the similarity between a short text segment and a target category cannot be estimated in this way. Recently, some researchers have tried to use Yahoo!'s directory as a knowledge source [Labrou and Finin 1999] for categorizing Web pages, but our approach is different from [Labrou and Finin 1999] in that, instead of depending on the rich information available from Yahoo!'s categorized pages, we only use Yahoo!'s category names and topic structure. In other words, our approach is more flexible and can be adapted to other hierarchies.

*Web Mining and Web Corpus*: Our research is related to text mining, which aims to discover knowledge in huge amounts of unstructured textual data from the Web. A variety of related subjects have been studied, such as the automatic extraction of terms or phrases [Feldman et al. 1997] and the discovery of rules for the extraction of specific information patterns [Soderland 1997]. Some works have also tried to exploit the Web as a corpus to solve NLP problems, such as word sense disambiguation [Agirre et al. 2000] and PP attachment [Volk 2001]. Different from these previous works, the proposed approach categorizes text segments by mining search-result snippets.

*Information Summarization*: Document summarization is a popular information summarization approach that aims at extracting important sentences as an abstract from single or multiple documents [Radev and McKeown 1998]. In addition, there were some approaches that use template-based information extraction techniques to assist with the extraction of important facts from documents. Organizing topic terms from documents into a topic hierarchy is another useful approach to obtaining an overview of the information contained in the documents [Sanderson and Croft 1999]. Different from these previous works, our approach focuses on categorizing text segments onto a well-formed thematic hierarchy without a pre-arranged training set for characterizing both text segments and thematic categories. We believe the proposed approach can also be integrated with previous works. For example, it can organize important sentences, facts or topic terms extracted by the previous approaches into a more comprehensive information structure. For database summarization, a number of approaches have been developed [Lee and Kim 1997], and most of them rely on the use of metadata information.

## 5. DISCUSSION AND CONCLUSION

In this work, we have presented a Web-based approach to generating thematic metadata and justified it empirically. The chief idea behind the proposed approach is to combine the structural information in the thematic hierarchy and Web resources. Because little human involvement is required, the proposed approach is flexible and can be easily adapted for other purposes. Although it still relies on a manual-given thematic hierarchy, the cost of creating such a thematic hierarchy is undoubtedly much less than labeling a large set of documents

Table XVI. Yahoo!'s Computer Science Experiment when the Corpus Size Increases

| # of Snippets | Top-1 | Top-2 | Top-3 | Top-4 | Top-5 |
|---|---|---|---|---|---|
| 100 | 0.4917 | 0.6346 | 0.6943 | 0.7242 | 0.7545 |
| 200 | 0.4850 | 0.6213 | 0.6910 | 0.7243 | 0.7409 |
| 400 | 0.4518 | 0.5880 | 0.6545 | 0.6910 | 0.7043 |
| 600 | 0.4219 | 0.5647 | 0.6445 | 0.6678 | 0.6810 |
| 800 | 0.4219 | 0.5083 | 0.5648 | 0.6047 | 0.6146 |

Approach 1. $N_{\max} = 100$. Search Engine: Google.

to train the needed classifiers; moreover, due to its use of Web corpora, the proposed approach is immune to the often-encountered problems of scalability and portability.

As we have discussed in the preceding section, many works have attempted to deal with problems using unlabeled data downloaded from the Web. This may leave the impression that there is not too much difference between the proposed approach in methodological aspect. We emphasize this is in fact not the case. The act of downloading Web documents, in the proposed approach serves not only as "enlarging" the training corpus.

Table XVI lists the inclusion rates of Computer Science experiment designed in Section 3.1 when the size of training corpora increased. It can be observed that the performance *did not* ameliorate with the size of the training corpora. A probable reason of this phenomenon is that the lowly-ranked snippets contain much more noise, thus hurting the accuracy. Obviously, downloading Web documents indiscriminately does not ensure success in training. The reason that HCQF can get better result is rather its exploiting structural information contained in thematic hierarchies.

An interesting extension of our approach would be to generate thematic metadata with fewer primitive metadata values (text segments). The relationship between text segments and the thematic metadata needs not be an one-to-one mapping: "Massachusetts" is created by categorizing "boston"; "Linguists" by "Noam Chomsky". Actually, a single text segment may create multiple thematic metadata. Given a single "People: Noam Chomsky," we can categorize it onto several sets of categories, such as a set of universities and a set of CS research fields (as described in Section 3.1), thereby deriving thematic metadata about where Chomsky taught and the field in which he was specialized.

## REFERENCES

AGIRRE, E., ANSA, O., HOVY, E., AND MARTINEZ, D. 2000. Enriching very large ontologies using the WWW. In *Proceedings of ECAI 2000 Workshop on Ontology Learning*.

COHEN, W. AND SINGER, Y. 1996. Context-sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, Eds. ACM Press, New York, US, Zürich, CH. 307–315.

DASARATHY, B. V. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. McGraw-Hill Computer Science IEEE Computer Society Press, Los Alamitos, California.

ERDMANN, M., MAEDCHE, A., SCHNURR, H., AND STAAB, S. 2000. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, Luxembourg*, August 2000.

FELDMAN, R., AUMANN, Y., AMIR, A., KLOESGEN, W., AND ZILBERSTIEN, A. 1997. Maximal association rules: A new tool for mining for keyword co-occurrences in document collections. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*. 167–170.

GAO, J.-F., LI, M., AND HUANG, C.-N. 2003. Improved source-channel models for chinese word segmentation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

GILLILAND-SWETLAND, A. J. 2000. "setting the stage: Defining metadata" in introduction to metadata: Pathways to digital information. Available online http://www.getty.edu/research/institute/standards/intrometadata/index.html.

JENKINS, C., JACKSON, M., BURDEN, P., AND WALLIS, J. 1999. Automatic RDF metadata generation for resource discovery. In *World Wide Web Conference*. 1305–1320.

KOSALA, R. AND BLOCKEEL, H. 2000. Web mining research: A survey. *SIGKDD: SIGKDD Explorations 2*, 1–15.

LABROU, Y. AND FININ, T. W. 1999. Yahoo! as an ontology: Using yahoo! categories to describe documents. In *ACM Conference on Information and Knowledge Management*. 180–187.

LEE, D. AND KIM, M. 1997. Database summarization using fuzzy ISA hierarchies. *IEEE Trans. Syst. Man Cybern. 27*, 1, 68–78.

MANNING, C. D. AND SCHÜTZE, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

MCCALLUM, A., FREITAG, D., AND PEREIRA, F. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA. 591–598.

RADEV, D. R. AND MCKEOWN, K. 1998. Generating natural language summaries from multiple on-line sources. *Computat. Linguist. 24*, 3, 469–500.

RDF. 1997. Available at http://www.w3.org/RDF/.

SALTON, G. AND BUCKLEY, C. 1988. Term weighting approaches in automatic text retrieval. *Inform. Process. Mgmt 24*, 513–523.

SANDERSON, M. AND CROFT, W. B. 1999. Deriving concept hierarchies from text. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 206–213.

SHLOMO, A., DAGAN, I., AND KRYMOLOWSKY, Y. 1998. Memory-based learning of shallow natural language patterns. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*.

SODERLAND, S. 1997. Learning to extract text-based information from the world wide web. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. 251–254.

STAAB, S., MAEDCHE, A., AND HANDSCHUH, S. 2001. An annotation framework for the semantic web. In *Proceedings of the First Workshop on Multimedia Annotation*, Tokyo, Japan, January 30–31, 2001.

UCHIMOTO, K., MA, Q., MURATA, M., OZAKU, H., AND ISAHARA, H. 2000. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

VOLK, M. 2001. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics*.

WKD. 2003. Available at http://WKD.iis.sinica.edu.tw/talipExperiment/.

YANG, Y. 2001. A study on thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 137–145.

Received July 2003; revised April 2004; accepted June 2004