

# Automatic Training Corpora Acquisition through Web Mining

Chien-Chung Huang  
Dartmouth College  
Hanover, New Hampshire  
villars@iis.sinica.edu.tw

Kuan-Ming Lin  
Duke University  
Durham, North Carolina  
km@cs.duke.edu

Lee-Feng Chien  
Academia Sinica  
Taipei, Taiwan  
lfchien@iis.sinica.edu.tw

## Abstract

*Text classification is a task having been extensively studied for decades. However, most previous work pre-assumes the existence of explicitly-labeled corpora. In this study, we focus on the issue of automatic corpora acquisition. We propose an Web-based mining approach to collect necessary corpora, which can be greatly useful to both common users and system designers. Moreover, the proposed technique can also be incorporated with existing classification techniques to further boost classifier performance.*

*It has been shown that the concept of the class can be captured by the class name and its associated terms [10]. In this work, we aim at analyzing Web-retrieved documents to discover the associated terms, which are further utilized to collect more training corpora. Working iteratively, the proposed approach can acquire training corpora of high quality. We give empirical evidence that the classifiers thus created have promising accuracy. In sum, the convenience and efficiency of the proposed approach, along with the new perspective on the issue of corpora acquisition, are the primary contributions of this work.*

## 1 Introduction

Text classification is an extensively studied problem with a long history [20, 17]. Most works in the literature focus on modeling and classification algorithms and implicitly assume that training corpora always can be organized.

In this work, we focus instead on corpora acquisition. This issue, though less studied in the literature, deserves investigation for two reasons. First, if the whole process of corpus collecting and labeling can be fully automated, the human involvement in the task of text classification can be much reduced: common users can easily create classifiers they personally defined; designers have more options in designing advanced applications and systems. Second, if there exist mechanisms able to fetch training corpora of good quality, then these mechanisms can be integrated with

many existing models and classification algorithms to further enhance the classifier performance.

In recent years, with the tremendous increase in digital documents, people began to use unsupervised corpora to reinforce or even substitute manually-labeled corpora [14, 13, 9]. [14, 13] only use a small set of labeled data (which can be labeled documents [14], or some manually-assigned keywords [13]); they use a bootstrapping process to label the unlabeled documents and re-train the classifier. [21] develops SVMC, a variant of the intensively studied SVM, that only uses a subset of positive examples.

The Web contains inexhaustible resources and has been exploited to solve various problems [4, 3, 7, 8, 5]. For text classification, [10] suggests a possible way of utilizing unsupervised Web corpora. It assumes that, instead of labeled corpora, only a manually-defined topic hierarchy is given (*Yahoo!*'s directory being a prototypical example); it then exploits the structural information hidden in the topic hierarchy, combined with Web search engines, to collect Web resources as corpora to train all classes in the hierarchy.

Inspired by [10], we propose a new approach to collect training corpora. No matter a topic hierarchy or a flat set of classes are given, by analyzing Web-retrieved snippets, we discover important associated terms that co-jointly capture the concept of the classes. These associated terms are further utilized to collect more training corpora. Working iteratively, the proposed approach greatly boost the classifier performance.

The rest of the paper is organized as follows. In Section 2 and Section 3, the proposed approach and the experimental results are given; some related work are discussed in Section 4 and the conclusion is drawn in Section 5.

## 2 The Approach

We first describe some technical fundamentals and then develop the approach progressively. The classic Vector Space Model is employed to model texts. After removing stop-words, each document is treated as a bag of word features and is projected to a point in the feature vector space.

The weight of each term is determined by the tf-idf formula. The distance of two documents is computed by the cosine angle between them.

## 2.1 First Attempt: using class names

Given a set of classes, the goal is to acquire suitable corpora to train each of them. We suppose that posing the class name as queries to Web search engines can get documents that describe the concept of the classes, which can consequently be treated as the training corpora.

To investigate the validity of this idea, we designed the following experiment. In *Yahoo!*'s directory, there was an enumeration of scientists of ten disciplines (Science/People/), including "physicists," "mathematicians" and so on. We wanted to train a classifier composed of these scientist classes. For each class, we submitted its name to the search engine<sup>1</sup> and then collected the first 100 result-snippets to train the class.

To evaluate the performance of the created classifier, we organized totally 211 biographies of scientists listed in *Wikipedia* [1] and classified them. Table 1 lists the results. Top-1 inclusion rate<sup>2</sup> in this experiment was as high as 52%.

**Table 1. Top 1-5 inclusion rates for classifying 211 scientist biographies from *Wikipedia*. # of Classes = 10.**

Top-1	Top-2	Top-3	Top-4	Top-5
.5188	.6493	.7299	.8098	.8343

**Remark 1** *This small experiment hints at the power of using Web snippets to train the classifiers. Though quite simple in conception and implementation, the accuracy the Web-based classifiers achieve is promising, not to mention its convenience and efficiency.*

It is natural to think about ways to refine this Web-based training approach. Among various alternatives, the most intuitive is to enlarge the number of snippets used for training. It is a commonly-held belief that the more sufficient the training corpus, the abler we are to overcome the problem of data spareness. We accordingly re-conducted experiment following this belief, i.e., we increased the number of training snippets and re-classified the testing *Wikipedia* articles. Table 2 lists the results. To our disappointment, the performance of the classifiers did not upgrade markedly with the number of training snippets. On the contrary, after a certain point (snippet size = 400), the performance dropped noticeably. A probable reason is that when using more snippets, more noise was also unavoidably taken in, thus impairing the quality of the training corpus. Apparently, to bring

<sup>1</sup>For all the experiments reported in this work, the search engine used was invariably Google.

<sup>2</sup>Top- $n$  inclusion rate is the metric we adopt to evaluate classification accuracy throughout this work. It means the percentage of test documents whose highly-ranked  $n$  candidate classes contain the correct class.

about better performance of the classifiers, we need better techniques to collect the needed corpora.

## 2.2 Second Attempt: HCQF

Also aiming at classification using Web-retrieved snippets, [10] proposes a technique called HCQF (Hier-Concept-Query-Formulation), which offers some insights on how to collect training corpora of better quality. We sketch its idea as follows. Instead of training a flat set of classes, [10] assumes that a topic hierarchy (taxonomy) is given and tries to organize the training corpora for all the classes in the hierarchy. To boost the accuracy, [10] made some refinements to HCQF. Highlights of its features are:

- When the class name is to be submitted to search engines, it has to be constrained by its ancestor classes.
- The training corpus of the sub-level classes can be used to enrich the corpus of the parent class.

For illustration of using HCQF, when collecting the training corpus for the class "Pierre de Fermat," a child class of "Mathematicians," we have to use the boolean expression of the two names to compose the query. Moreover, when organizing the corpus of "Mathematicians," we can use the corpus of not only "Mathematicians" itself but also those of its sub-level classes (one of which is "Pierre de Fermat").

**Table 2. Top 1 inclusion rate for classifying scientist biographies from *Wikipedia*. Per 100 snippets were treated as an article.**

# of Snippets	100	200	300	400	500	600	700	800
Top-1 inclusion rate	.5188	.5260	.5213	.5355	.4787	.4076	.3981	.3697

To facilitate further discussions, we introduce some notations. Given a set of user-defined classes  $\mathcal{U}$ , for each class  $C^* \in \mathcal{U}$ , after applying HCQF, its training corpora is then composed of a set of training corpora  $C^0 = \{c_0, c_1 \dots c_{|C^0|-1}\}$ , among which  $c_0$  corresponds to the training corpus retrieved by using  $C^*$  as query<sup>3</sup>, while  $c_1, c_2$  and so on the corpora retrieved by using the boolean expression of  $C^*$  and its sub-level class  $C_{SL_j}^*$ . In current stage, we omit to explain the meaning of the superscript of  $C^0$  lest our focus should be diverted.

Based on the preceding idea, we re-conducted the experiment. In *Yahoo!*'s Scientists directory, there were sets of sub-level classes belonging to each type of scientists, e.g. "Albert Einstein" was a child class of "Physicists" and "Marie Curie" a child class of "Chemists" and so on. For each class, we used the first 5 sub-level classes (in alphabetical order) and made them a three-level deep topic hierarchy

<sup>3</sup>For brevity of presentation, we use the notation  $C^*$  in two senses. It may mean the class  $C$  itself or the name of the class  $C$ .

and then applied the technique HCQF to it. For class  $C^*$ , its corpus  $C^0 = \{c_0, c_1, \dots, c_5\}$  can be converted to a set of points in the feature vector space and their centroid  $\bar{C}^0$  can be thus computed. We then classified Web sites depending on their relative distance to each centroid. The top-1 inclusion rate now jumped to 64%, a marked improvement compared to our First Attempt. (Detailed figures showing all top-3 accuracy can be seen in Section 3.)

**Remark 2** *The above experiment suggests that, to better the quality of training corpus, a more effective way is to use the associated terms to acquire more corpus. One may argue that in practice it cannot be expected that there always exists a topic hierarchy of classes to be exploited. Nevertheless, HCQF points a new direction on how to acquire more suitable training corpora, i.e., to capture the class concept, the associated terms are helpful. We shall fully develop our technique following this line of thinking.*

### 2.3 Third Attempt: discovering associated terms

Given a class  $C^* \in \mathcal{U}$ , we denote  $C^0 = \{c_0, \dots, c_{|C^0|-1}\}$  as the Web-retrieved corpus that HCQF organized. (If, as in the First Attempt, we are only given a flat set of classes, then  $C^0 = \{c_0\}$ , a special case.) Analyzing the corpus  $C^0$ , we aim to discover some associated terms of  $C^*$ , so that we can use the “query expansion” technique, i.e., sending the boolean expression  $C^*$  and the associated terms to acquire another set of training corpora  $\hat{C}^0 = \{c_{|C^0|}, c_{|C^0|+1}, \dots\}$ , adding them to  $C^0$ , and obtaining a richer set of training corpora for  $C^*$ . We apply the above procedure iteratively. In other words, in each round  $a$ ,

$$C^a = C^{a-1} \cup \hat{C}^{a-1}.$$

The problem, then, is given  $C^a$ , how can we decide which terms are really associated. A tempting idea is to choose high frequency terms or those terms often co-occurred  $C^*$ . However, many of these terms may be very common words and abound in the corpus of all classes. Using these terms to collect new training corpora might make the training corpora of each class “resemble” one another. And this obviously runs against the intuition as the classes become less and less distinct.

The above consideration leads us to design the following formula. Given an unknown term  $u$ , it cannot be a proper candidate term unless

$$\frac{P(u|C^*)}{P(u|D^*)} > \nu Sim(C^*, D^*)^\mu, D^* \neq C^*, \forall D^* \in \mathcal{U}. \quad (1)$$

$D^*$  being some class other than  $C^*$ , the intuition behind Formula (1) is that the unknown term  $u$  must be very unique in the context of *all* classes. If it occurs frequently in the corpora of two (or more) classes, then it is probably a common word hence not suitable to be deemed as an associated

term of  $C^*$ . On the other hand, for those low-frequency terms, if they occur very rarely in the corpora of *other* classes, then they still have a chance of being “saved” to be incorporated into  $C^*$ .

For the right-hand-side of the equation, instead of using a fixed value, we use a similarity function  $Sim(C^*, D^*)$  to decide the threshold, because we have to take into account the inherent similarity of two sets of corpora. For two very dissimilar classes, say “Aviators” and “Mathematicians,” we should use a stricter threshold to filter out unsuitable terms, while if two similar classes, say “Mathematicians” and “Physicists,” we should lower the threshold, otherwise, it is possible that the two classes cancel out all associated terms of each other.

In computing  $Sim(C^*, D^*)$ , we use the cosine angle of their respective centroids, i.e.,  $Sim(C^*, D^*) = \cos(\bar{C}^a, \bar{D}^a)$ .

We now shift our focus to deal with the problem of how to approximate  $P(u|C^*)$ , the probability of sampling a term from class  $C^*$ . In round  $a$ , we know that  $C^a = \{c_0, c_1, \dots, c_{|C^a|-1}\}$  so we can decompose  $P(u|C^*)$  into the joint probability of  $u$  and  $C^a$ :

$$P(u|C^*) = \sum_{c_i \in C^a} P(u, c_i|C^*).$$

Using Bayes’ rule, we derive:

$$\sum_{c_i \in C^a} P(u, c_i|C^*) = \frac{\sum_{c_i \in C^a} P(u, C^*|c_i)P(c_i)}{P(C^*)}.$$

We assume that given a training corpus  $c_i$ , term  $u$  and class  $C^*$  are conditionally independent, thus,

$$\begin{aligned} \frac{\sum_{c_i \in C^a} P(u, C^*|c_i)P(c_i)}{P(C^*)} &= \sum_{c_i \in C^a} \frac{P(c_i)P(u|c_i)P(C^*|c_i)}{P(C^*)} \\ &= \sum_{c_i \in C^a} P(u|c_i)P(c_i|C^*). \end{aligned}$$

The outcome this series of derivations is a formula quite conforming to intuition: The probability of a term sampled from a class is the summation of the probability of a training corpus being sampled from the class multiplying the probability of sampling the term from the training corpus.

$P(u|c_i)$  can be obtained by simply dividing the number of  $u$ ’s occurrence in  $c_i$  with the total number of terms in  $c_i$ . However, it is possible that  $u$  does not exist in  $c_i$ , therefore, some smoothing is needed. Suppose  $T^a$  is the collection of all terms in all training corpora in round  $a$  (note  $T^a$  is a multiset) and  $n(u, c_i)$  the number of terms  $u$  in  $c_i$ , then,

$$P(u|c_i) = \frac{n(u, c_i) + 1}{|c_i| + |T^a|}.$$

Note that  $\sum_{u \in T^a} P(u|c_i) = 1$  is satisfied.

We are in a more difficult situation when dealing with  $P(c_i|C^*)$ . Confronted with an abstract class  $C^*$ , we have to resort to some more concrete substitute to “materialize” it. Considering that  $C^0$  is the training corpus acquired by using manually defined keywords, we assume that, comparatively, they can capture the concept of class  $C^*$  to a certain degree. In other words, we attempt to approximate  $P(c_i|C^*)$  by  $P(c_i|C^0)$ . After normalizing, we get:

$$P(c_i|C^*) = \frac{P(c_i|C^0)}{\sum_{c_j \in C^a} P(c_j|C^0)}.$$

Note that only the corpora in  $C^0$  are collected using manually defined keywords, all others are organized by automatic mechanisms and we are less confident in them. We use a uniform distribution to estimate the former; to estimate the latter, we utilize the document frequency returned by Web search engines. Denoting the document frequency of a query as  $df$ . Suppose  $c_i$  is derived from the boolean expression of  $C^*$  and term  $u_i$ , then,

$$P(c_i|C^0) = \begin{cases} 1/|C^0| & \text{if } i \leq |C^0| - 1, \\ \frac{df(C^*, u_i)}{df(C^*)} & \text{otherwise.} \end{cases} \quad (2)$$

Even if an unknown term  $u$  has passed the examination of Formula (1), it is still possible that it is a too common word, which contributes little, or even may be detrimental to the quality of the corpora of  $C^*$ . Again using the  $df$  value returned by the search engine, we decree that term  $u$  must also satisfy the following Jaccard metric:

$$\frac{df(u, C^*)}{df(u) + df(C^*) - df(u, C^*)} > \varphi \quad (3)$$

The whole algorithmic procedure is shown in Figure 1. As we shall discuss in the next section, the extracted terms not only help to boost the classifier accuracy, but they are themselves usually very meaningful terms. They can be thought of as the by-product of the proposed approach.

### 3 Experiments

We conducted two experiments to evaluate the proposed approach. One was, as previously discussed, the experiment concerning scientist classification; the other one was about classifying academic computer science papers. When sending queries to the search engine, we collected 100 snippets as an article in the corpora.  $\nu$ ,  $\mu$  in Formula (1) and  $\varphi$  in Formula (3) were set to 6.0, 0.1 and 0.015.<sup>4</sup> For comparison, we designed a number of scenarios.

<sup>4</sup>Here we only present one possible parameter set; in general, parameter selections can done through a variety of cross-validation approaches.

**Input:**  $\mathcal{U}$   
 $\mathcal{U}$ : user-defined classes (and their sub-level classes)

```

1: for all  $C^* \in \mathcal{U}$  do
2:    $c_0 \leftarrow$  send  $C^*$  (to the Web search engine)
3:    $C^0 \leftarrow C^0 \cup c_0$ 
4:    $T^0 \leftarrow T^0 \cup \{\text{all terms in } c_0\}$ 
5:   for all  $C_{SL_j}^*$  do {each of which is a sub-level class of  $C^*$ }
6:      $c_j \leftarrow$  send boolean expression of  $C^*$  and  $C_{SL_j}^*$ 
7:      $C^0 \leftarrow C^0 \cup c_j$ 
8:      $T^0 \leftarrow T^0 \cup \{\text{all terms in } c_j\}$ 
9: for  $a=0$  to Infinity do
10:   $T^{a+1} \leftarrow T^a$ 
11:  for all  $C^* \in \mathcal{U}$  do
12:     $C^{a+1} \leftarrow C^a$ 
13:    for all  $u \in T^a$  do
14:      for all  $D^* \in \mathcal{U}, D^* \neq C^*$  do
15:        if  $u$  passes the threshold of Formulae (1) and (3) then
16:           $c_{temp} \leftarrow$  send boolean expression of  $u$  and  $C^*$ 
17:           $C^{a+1} \leftarrow C^{a+1} \cup c_{temp}$ 
18:           $T^{a+1} \leftarrow T^{a+1} \cup \{\text{all terms in } c_{temp}\}$ 
19: if  $\forall C^* \in \mathcal{U}, C^{a+1} = C^a$  then
20:    $\forall C^* \in \mathcal{U}, \text{Output } C^i$ 
21:   Stop

```

**Figure 1.** An algorithmic procedure describing the proposed approach.

- Scenario  $X$ : The users specified a set of classes. We simply sent the class names to search engines (as discussed in the First Attempt) to organize the training corpora.
- Scenario  $X^+$ : The users specified a set of classes. We applied the proposed approach discussed in the Third Attempt to organizing the corpora.
- Scenario  $Y$ : The users specified a topic hierarchy. We applied HCQF as discussed in the Second Attempt.
- Scenario  $Y^+$ : The users specified a topic hierarchy. First applying HCQF, we then continued to use the technique in the Third Attempt to discover associated terms and to get more corpora.
- Scenario  $Z$ : The users specified the classes and also offered labeled training corpora. Scenario  $Z$  and Scenarios  $X, X^+, Y, Y^+$  could be deemed as the comparison between supervised and unsupervised training.

#### 3.1 Scientist Classes Experiment

In this experiment, the setting of Scenario  $X, X^+, Y^+$  and  $Y^+$  were as described in the preceding section. As to Scenario  $Z$ , we organized the site-list (including the descriptions) of *Yahoo!*'s directory as the labeled corpora. All the sites appearing in the same Web page are collectively treated as an article. We crawled *Yahoo!*'s Scientist directory four-level deep and collected a total of 353 articles.

**Table 3. Associated terms extracted for each class in Scenario  $X^+$  of Scientist experiment.**

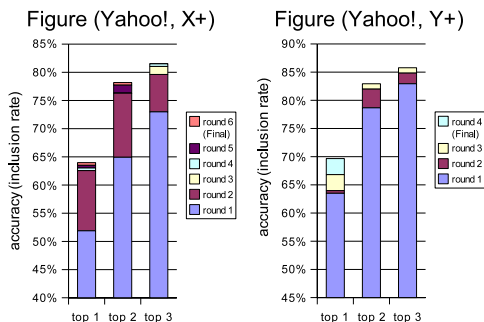
Astronauts	Apollo, cosmonauts, landing, Lunar, NASA, Cernan, rocket, shuttle, UFOs, manned, Soyuz, ISS, spacecraft, capsule
Astronomers	skies, astronomy, comet, Milky, craters, Jupiter, planets, discoverer, observatory, comets, planetary, earliest, clouds, Herschel, astrophysics, astrophysicists, nebula, extrasolar, telescope, universe, galaxies
Aviators	pilots
Biologists	watershed, obvious, genetic, habitat, taxonomists, disturbed, botanical, basin, collected, evolutionary, trout, fisheries, wildlife
Earth Scientists	metals, geography, fossils, mineral, influences, sciences, crust, geology, mining, meteorology, developments, measuring
Chemists	Nobel, ACS, nontraditional, analytical, biological, medicinal, pharmaceutical, chemistry, organic, synthetic, compounds
Engineers & Inventors	
Mathematicians	MCS, algebra, philosophers, mathematics, DCS, Mactutor
Physicists	particle, physics, Bohr, quantum, theoretical, mechanics
Primatologists	primate, primatology, chimpanzees, anthropologists, primatological

**Table 4. Scientist experiment results and related information.**

Scenario	# of (distinct) words	Top-1	Top-2	Top-3
$X$	15811 (4783)	.5188	.6493	.7299
$X^+$	259740 (30024)	.6351	.7677	.8104
$Y$	102763 (17021)	.6351	.7867	.8294
$Y^+$	308440 (33868)	.6967	.8151	.8530
$Z$	19669 (4436)	.4663	.6380	.7301

Table 4 lists the result of the five scenarios. The gain in performance from Scenario  $X$  to Scenario  $X^+$  was obvious, implying the proposed approach could help acquire training corpora of good quality. Figure 2 displays the more detailed results of Scenario  $X^+$ , listing how the performance of the classifier had improved in successive rounds. Table 3 lists the extracted associated terms in Scenario  $X^+$ , which can be observed to be usually relevant to the classes. One might worry that some improperly extracted terms might have further bad influence on subsequent extracted terms (since the proposed approach is iterative). However, if the extracted term was really irrelevant to the class, Formula (2) ensured that its contribution be small. On the other hand, there are

**Figure 2. Yahoo! Results.**



also terms in Table 3 that may appear irrelevant but, if carefully examined, turn out to be meaningful. For example, for “Mathematicians” class, MCS is a division of Argonne National Laboratory; DCS is a website comprising detailed mathematicians materials. Another example is that, for the “Astronauts” class, ISS is the abbreviation of “International

Space Station.”

It is worthwhile to point out that the accuracy in  $X^+$  and in  $Y$  were comparable. This implies that we could organize training corpora of as good quality as those organized using human-defined topic hierarchy, even though the extracted terms might not be as precise and as meaningful as those specified by humans.

In scenario  $Y^+$ , the proposed approach also could help upgrade the classifier performance as shown in Figure 2, though the extent of improvement was not as manifest as in  $X^+$ —it is often harder to improve a thing with already-high quality.

It is also interesting to observe that the results in  $X$  and  $Z$  were very close, even the sizes of their training corpora were similar. Only using the class name to get unlabelled corpora can have comparable results with those manual-labelled corpora. This seemed to testify to the potential of using Web-retrieved documents. Considering the proposed approach only needs the minimal manual labor, we think it may be embedded to many advanced applications.

### 3.2 CiteSeer Paper Experiment

We conducted another experiment using the classes defined by the famous CiteSeer website [2], which also offers a browsable topic hierarchy and lists of academic papers. Among the 17 classes (along with their sub-level classes), we dropped the “World Wide Web” and “Applications” classes as they overlapped with too many other classes. For each of the 15 classes, we collected the first 40 papers listed and extracted their abstracts as our testing documents in Scenario  $X$ ,  $X^+$ ,  $Y$  and  $Y^+$ . As to Scenario  $Z$ , unlike the previous experiment where open test instances exist, we use 5-fold cross validation to evaluate. Table 5 lists the detailed information about manually-defined terms and automatically-extracted associated terms in Scenario  $Y^+$ . Table 6 lists the result of the five scenarios, and Figure 3 depicts the accuracy progress in Scenarios  $X^+$  and  $Y^+$ . A similar conclusion could be reached with the previous experiment. The proposed approach did help to boost the accuracy of the classifiers in Scenarios  $X^+$  and  $Y^+$ . However, the overall results in all scenarios were not as impressive as

**Table 5. Sub-level classes and extracted terms for the Computer Science experiment in Scenario  $Y^+$ .**

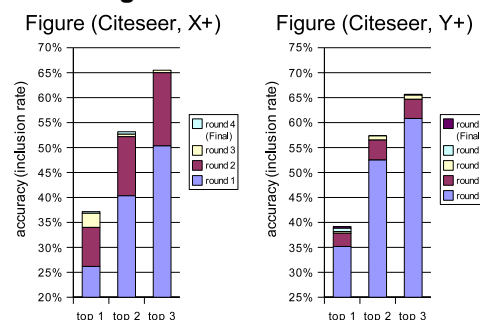
Class	Sub-level Classes (Manually-Assigned Keywords)	Extracted Associated Terms
Agents	Architecture, Assistant Agent, BDI, Mobile Agents , Multi-agent Systems , Synthetic Agents	rational, autonomous, AAMAS
Architecture	Clusters, Distributed, Parallel	pipelined, quantitative, SIGARCH
Artificial Intelligence	Expert Systems, Knowledge Representation, Natural Language Processing, Optimization, Planning, Robotics	AI
Compression	Audio, Text, Video	wavelet, codec, GIF, encoding, codecs, formats, compressor, coding, lossy, lzw, gzip, png, lossless, compressed, jpeg, Ogg
Databases	Concurrency, Data Warehousing, Deductive, Object-oriented	spatial
Hardware	CISC, High Performance, Logic Design, Memory Structures, Micro-programming, RISC, Storage, VLSI	programmable, circuits
Human Computer Interaction	Collaboration, Graphics, Interface Design, Multimedia, Ubiquitous Computing, Virtual Reality, Wearable Computing, Workflow Systems	SIGCHI, interruptions, HCI, CSCW, interfaces
Information Retrieval	Classification, Digital Libraries, Extraction, Filtering, Metasearch, Retrieval, Search Engines, World Wide Web	retrieve, chemistry, SIGIR, IR
Machine Learning	Case-based Learning, Fuzzy Systems, Genetic Algorithms, Neural Networks, Pattern Recognition, Reinforcement Learning , Rule Based Systems , Vision	ICML
Networking	ATM, Internet, Local Area, Multicast, Protocols, Routing, TCP , Wide Area , Wireless	WANs, LANs, routers, CISCO, Fi, relay, ios, switches, ethernet, connect, hubs, gigabit, dial
Operating Systems	Clusters, Distributed, Fault Tolerance, Linux, Memory Management, Microkernel, Real-time, Unix, Windows	RTOS, solaris, realtime, posix
Programming	Compiler Design, Compiler Optimization , Functional , Java, LISP, Logic , Memory Management , Object-oriented , Open Source, Semantics	compilers, garbage, ICFP, EMACS, interpreters, contest
Security	Access Control, Encryption, Information Warfare, Intellectual Property Protection, Intrusion Detection	TOSEM, ICSE
Software Engineering	Data Structures, Parallelism, Randomized Algorithms	
Theory	Computational Complexity, Formal Languages, Logic , Quantum Computing, Theorem Proving	automata, theoretical, foundations, physics, math, discrete, combinatorics, probability

**Table 6. Computer Science experiment results and related information. Scenario  $Z$  is conducted in 5-fold cross validation.**

Scenario	# of (distinct) words	Top-1	Top-2	Top-3
$X$	25103 (5512)	.2617	.4033	.5033
$X^+$	365683(29842)	.3716	.5317	.6517
$Y$	218558 (21228)	.3517	.525	.6083
$Y^+$	599797(44019)	.3917	.5717	.655
$Z$	100413.6 (7504.8)	.3425	.5196	.6312

before. This was caused by two reasons: (a) The number of the target classes increased ( $10 \rightarrow 15$ ); and (b) Relatively speaking, the concept of the computer science classes were not very distinct. The retrieved documents concerning “Agent” must often have similar contents with documents concerning “Artificial Intelligence.” This naturally raised the difficulty of training these classes.

**Figure 3. Citeseer Results.**



## 4 Related Work

Using unlabelled corpora to train classifiers [14, 13, 9], as discussed in Section 1, is a topic draws much attention these days. Concerning treating the Web as a source to extract corpora, a summary work can be found in [11].

Some relevant studies on Internet knowledge discovery have appeared in the agent systems field. For an overview,

[12] outlined the components of intelligent agents for the Internet. An example is KAROKA [16], designed for scientific bibliography investigation. It uses Web pages sources to generate and to combine randomly keywords, then tries to find association rules. However, the Web page sources are directly from the topic hierarchy of search engines (e.g. *Yahoo!*), so the available corpora are still limited.

In using the associated terms of the classes to collect Web corpora, we have essentially applied the query expansion technique, though the goal in this study is quite different. For a summary article on query expansion, see [18]; more recent developments can be found in [15, 19, 6]. It seems there exists much possibility of incorporating the local text analysis of query expansion into the proposed approach to get more meaningful associated terms.

## 5 Conclusion and Future Work

In this work, we have proposed a Web-based approach that can train classifiers automatically. Unlike previous works depending on manually-labeled corpora, the approach needs only a few keywords to organize training corpora of rather good quality. In practice, the efficiency and the convenience are its main advantages; in theory, it opens up a new research avenue of text classification—corpora acquisition. We made the observation that, to capture the concept of a class, rather than simply enlarging the corpus size, a better strategy is to use its associated terms to approximate it. The whole approach was built following this observation. We are exploring how to integrate it with other existing text classification algorithms.

We believe that techniques in Natural Language Processing and Information Extraction can also be incorporated to further upgrade the classifier performance. For example, in the experiment, we indiscriminately treated the returned 100 snippets as an article of the training corpora. However, if we employ some linguist-aware filtering mechanisms, many noise snippets might be filtered out, and quality of training corpora could thus be improved. Also, we only considered uni-grams in extracting new associated terms, many bi/tri-gram (as shown in the case of the manually-defined keywords in Table 5) may also improve the quality of the resulting corpora. How to incorporate bi/tri-gram term extraction techniques to our technique might be a promising future research direction.

## References

- [1] Available at <http://en.wikipedia.org/wiki/>.
- [2] Available at <http://citeseer.ist.psu.edu/directory.html>.
- [3] E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the www. In *Proceedings of ECAI 2000 Workshop on Ontology Learning*, 2000.
- [4] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *Proceedings of the 11st International World Wide Web Conference*, pages 26–33, 2002.
- [5] D. W. C. Kwok, O. Etzioni. Scaling question answering to the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 150–161, 2001.
- [6] C. Carpineto, R. De Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.
- [7] W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference*, pages 307–315, Zürich, CH, 1996. ACM Press, New York, US.
- [8] R. Goldman and J. Widom. Wsq/dsq: A practical approach for combined querying of databases and the web. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 285–296, 2000.
- [9] J. H. H. Yu, C. Zhai. Text classification from positive and unlabeled documents. In *Proceedings of the 12th Annual International ACM Conference on Information and Knowledge Management*, pages 232–239, 2003.
- [10] C.-C. Huang, S.-L. Chuang, and L.-F. Chien. Liveclassifier: Creating hierarchical text classifiers through web corpora. In *Proceedings of the 10th International World Wide Web Conference*, pages 184–192, 2004.
- [11] A. Kilgarriff and G. Greffentette. Introduction to the special issue on web as corpus. *Computational Linguistics*, 29(3), 2003.
- [12] M. Klusch. Information agent technology for the internet: A survey. *Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration*, 36(3), 2001.
- [13] A. McCallum and K. Nigam. Text classification by bootstrapping with keywords. In *ACL Workshop for Unsupervised Learning in Natural Language Processing*, 1999.
- [14] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.
- [15] Y. Qui and H. Frei. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference*, pages 160–169, 1993.
- [16] D. Ramamonjisoa. Research topics discovery from www by agent system. In *International Conference on Advances in Infrastructure for e-Electronic, e-Business, e-Education, e-Science, e-Medicine on the Internet, paper number 13*, 2003.
- [17] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.
- [18] K. Sparck-Jones. Notes and references on early classification work. *SIGIR Forum*, 25(1):10–17, 1991.
- [19] J. Xu and W. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference*, pages 412–420, 1996.
- [20] Y. Yang. A study on thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference*, pages 137–145, 2001.
- [21] H. Yu. Svmc: Single-class classification with support vector machines. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.