# Study on Building a High-Quality Homepage Collection from the Web Considering Page Group Structures

**Yuxin WANG**

**DOCTOR OF PHILOSOPHY**

Department of Informatics
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies (SOKENDAI)

2006 (School Year)

September 2006

# Study on Building a High-Quality Homepage Collection from the Web Considering Page Group Structures


## Abstract

This disseration is devoted to investigate the method for building a high-quality homepage collection from the web efficiently by considering the page group structures. We mainly investigate in researchers' homepages and homepages of other categories partly.

A web page collection with a guaranteed high quality (i.e., high recall and high precision) is required for implementing high quality web-based information services. Building such a collection demands a large amount of human work, however, because of the diversity, vastness and sparseness of web pages. Even though many researchers have investigated methods for searching and classifying web pages, etc., most of the methods are best-effort types and pay no attention to quality assurance. We are therefore investigating a method for building a **homepage collection** efficiently while assuring a given high quality, with the expectation that the investigated method can be applicable to the collection of various categories of homepages.

This dissertation consists of seven chapters. Chapter 1 gives the introduction, and Chapter 2 presents the related work. Chapter 3 describes the objectives, the overall performance goal of the investigated system, and the scheme of the system. Chapters 4 and 5 discuss the two parts of our two-step-processing method in detail respectively. Chapter 6 discusses the method for reducing the processing cost of the system, and Chapter 7 concludes the dissertation by summarizing it and discussing future work.

**Chapter 3**, taking into account the enormous size of the real web, introduces a two-step-processing method comprising **rough filtering** and **accurate classification**. The former is for narrowing down the amount of candidate pages efficiently with the required high recall and the latter is for accurately classifying the candidate pages into three classes—assured positive, assured negative, and uncertain—while assuring the required recall and precision.

We present in detail the configuration, the experiments, and the evaluation of the rough filtering in **Chapter 4**. The rough filtering is a method for gathering researchers' homepages (or entry pages) by applying our original, simple, and effective local page group models exploiting the mutual relations between the structure and the content of a logical page group. It aims at narrowing down the candidates with a very high recall. First, **property-based keyword lists** that correspond to researchers' common properties are created and are grouped either as **organization-related** or **non-organization-related**. Next, four **page group models (PGMs)** taking into consideration the structure in an individual logical page group are introduced. **PGM_Od** models the out-linked pages in the same and lower directories, **PGM_Ou** models the out-linked pages in the upper directories, **PGM_I** models the in-linked pages in the same and the upper directories, and **PGM_U** models the site top and the directory entry pages in the same and the upper directories.

Based on the PGMs, the keywords are propagated to a potential entry page from its surrounding pages to compose a virtual entry page. Finally, the virtual entry pages that scored at least a threshold value are selected. Since the application of PGMs generally causes a lot of noises, we introduced four **modified PGMs** with two original techniques: the keywords are propagated based on PGM_Od only when the number of out-linked pages in the same and lower directories is less than a threshold value, and only the organization-related keywords are propagated based on other PGMs. The four modified PGMs are used in combination in order to utilize as many informative keywords as possible from the surrounding pages.

The effectiveness of the method is shown by comparing it with that of a single-page-based method through experiments using a 100GB web data set and a manually created sample data set. The experiment results show that the output pages from the rough filtering are less than 23% of the pages in the 100GB data set when the four modified PGMs are used in combination under a condition that the recall is more than 98%. Another experiment using a 1.36TB web data set with the same rough filtering configuration shows that the output pages are less than 15% of the pages in the corpus.

In **Chapter 5** we present in detail the configuration, the experiments, and the

evaluation of the accurate classification method. Using two types of component classifiers (a **recall-assured classifier** and a **precision-assured classifier)** in combination, we construct a **three-way classifier** that inputs the candidate pages output by the rough filtering and classifies them to three classes: **assured positive**, **assured negative**, and **uncertain**. The assured positive output assures the precision and the assured positive and uncertain output together assure the recall, so only the uncertain output needs to be manually assessed in order to assure the quality of the web data collection.

We first devise a feature set for building the high-performance component classifiers using Support Vector Machine (SVM). We use **textual features** obtained from each page and its surrounding pages. After the surrounding pages are grouped according to **connection types** (in-link, out-link, and directory entry) and **relative URL hierarchy** (same, upper, or lower in the directory hierarchy), an independent **feature subset** is generated from each group. Feature subsets are further concatenated conceptually to compose the **feature set** of a classifier. We use two types of textual features (**plain-text-based** and **tagged-text-based**). The classifier using only the plain-text-based features in each page alone is used as the baseline. Various feature sets are tested in the experiment using manually prepared sample data, and the classifiers are tuned by two methods, one *offset*-based and the other *c-j-option*-based. The results show that the performance obtained by using *c-j-option*-based **tuning method** is statistically significant at 95% confidence level. The F-measures of the baseline and the top two performed classifiers are 83.26%, 88.65%, and 88.58% and show that the proposed method is evidently effective.

To know the performances of the classifiers with the abovementioned feature sets in more general cases, we experimented with our method on the **Web->KB data set**, a test collection commonly used for the web page classification task. It contains seven categories and four of them—*course*, *faculty*, *project*, and *student*—are used for comparing the performance. The experiment results show that our method outperformed all seven of the previous methods in terms of macro-averaged F-measure. We can therefore conclude that our method performs fairly well and is applicable not only to researchers' homepages in Japanese but also to other categories of homepages

in other languages.

By tuning the well-performing classifiers independently, we then build a recall-assured classifier and a precision-assured classifier and compose a three-way classifier by using them in combination. We estimated the numbers of the pages to be manually assessed for the required precision/recall at 99.5%/98%, 99%/95%, and 98%/90%, using the output pages from a 100GB data set through the rough filtering. The results show that the **manual assessment cost** can be reduced, compared to the baseline, down to 77.6%, 57.3%, and 51.8%, respectively. We analyzed classification result examples, and the results show the effectiveness of the classifiers.

In **Chapter 6** the **cascaded structure** of the recall-assured classifiers, used in combination with the rough filtering, is proposed for reducing the computer processing cost. Estimation on the numbers of pages requiring feature extraction in the accurate classification shows that the **computer processing cost** can be reduced down to 27.5% for the 100GB data set and 18.3% for the 1.36TB data set.

In **Chapter 7** we summarize our contributions. One of our unique contributions is that we pointed out the importance of assuring the quality of web page collection and proposed a framework for doing so. Another is that we introduced an idea of local page group models (PGMs) and demonstrated its effective uses for filtering and classifying web pages.

We first presented a realistic framework for building a high-quality web page collection with a two-step process, composing the rough filtering followed by the accurate classification, in order to reduce the processing cost. In the rough filtering we contributed two original key techniques used in the modified PGMs to reduce the irrelevant keywords to be propagated. One is to introduce a threshold on the number of out-linked pages in the same and lower directories, and the other is to introduce keyword list types and propagate only the organization-related keyword lists from the upper directories. In the accurate classification we contributed not only a original method for exploiting features from the surrounding pages and concatenating the features independently to improve web page classification performance but also a way to use a recall-assured classifier and a precision-assured classifier in combination as a three-way classifier in order to reduce the amount of pages requiring manual

assessment under the given quality constraints.

We also discuss the future work: finding a more systematic way for modifying the property set and property-based keywords for the rough filtering, investigating ways to estimate the likelihood of the component pages and incorporate them for the accurate classification, and further utilizing the information from the homepage collection for practical applications.

# Acknowledgements

First and foremost, I would like to express my profound gratitude to my research supervisor, Professor Keizo Oyama, for leading me toward the completion of this dissertation. This work could never have been completed without his guidance, encouragement, enlightenment, and constant support. I would also like to express my thanks and appreciations to my supervisors, Professor Akiko Aizawa and Professor Atsuhiro Takasu, as well as Professor Jun Adachi, Professor Noriko Kando, and Professor Hideaki Takeda, for their advices and comments.

This Ph.D. work was financially supported by the Ministry of Education, Culture, Sports, Science and Technology, and the NW100G-01 and NW1000G-04 document data sets were used with the permission of the National Institute of Informatics. I would like to thank all the people in these organizations.

This dissertation is specially dedicated to my family, without whose continuous encouragement, its completion would not have been possible.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivations and Objectives

Electronic publishing technology has greatly reduced the amount of human effort needed to provide High-quality scholarly information services, and the World Wide Web is becoming a more and more important source of information for such services [1] [2] [3] [4] [5]. These services thus need to be based on web page collections that not only contain a high percentage of all the relevant documents (i.e., that are high recall collections) but also contain a high percentage of relevant documents (i.e., that are also high precision collections). Assembling such high-quality web page collections, however, takes a large amount of human work because of the diversity in style, granularity, and structure of web pages, the vastness of the web data, and the sparseness of relevant pages.

Collecting research information from the Web is very important [6] [7] [8], and search engines like Google Scholar[1] and database services such as CiteSeer[2], ReaD[3], and Web of Knowledge[4] provide information on research papers through search functions using keywords, author names, citations, etc. They provide little or no support, however, for the identification of the authors. Although these services

[1]Google Scholar. http://scholar.google.com/.

[2]CiteSeer. http://citeseer.ist.psu.edu/.

[3]ReaD. http://read.jst.go.jp/.

[4]ISI Web of Knowledge. http://portal.isiknowledge.com/.

are effective even without author identification, they will provide richer information both on papers and researchers if a high-quality information resource regarding researchers is available.

Information on researchers itself is a useful research resource, and some application systems [9] [10] [11] [12] [13] would be more effective if information on all the researchers were made available. Because such information could be obtained by assessing the researchers' homepages, this research focuses on the problem of collecting researchers' homepages from the Web. A guarantee-type information service like CiNii[5] could be extended by considering a collection of researchers' homepages if that collection were both complete (high recall) and accurate (high precision), but it is very difficult to guarantee the completeness and the accuracy of web-based information. Almost all of the related applications have therefore been only best-effort types, especially in terms of recall. We are thus trying to devise a method for comprehensively building a quality-guaranteed homepage collection—a collection with both high recall and high precision—that can be used by web-based applications.

It is not easy to collect homepages by using existing methods [14]. One problem is that there is no consistent style for a specific class of entities. The information of an entity can be presented on a single page or a set of pages that constitutes a logical page group. Another problem is that the granularity of the information in a web page varies from page to page. We are therefore investigating a method for building a homepage collection by exploiting web-based features.

Although some methods that take into consideration only the contents of single pages can collect homepages in single pages [15] to a certain degree, they perform rather poorly in the case of collecting homepages in logical page groups. Although methods that take into consideration the global link structure (e.g., in/out-link references and anchor texts across web sites) are quite effective for collecting popular researchers' homepages [16] [17], they are of almost no use in collecting homepages rarely referred to by other pages. Some methods exploit a variety of information related to a target page—e.g., textual content [18] [19] [20], html tag [21] [22], page

[5]CiNii is a research paper navigation service provided by the National Institute of Informatics. For details, see http://ci.nii.ac.jp/.

structure [23] [24] [25] [26] [27], page layout [28] [29] [30] [31] [32], URL [32] [33] [34] [35], directory structure [36] [37] [38] [39], anchor text [40] [41] [42] [43], and hyperlink structure [44] [45] [46] [47]—but only a few of them exploit information contained in the component pages [48] [49] [50] and less in combination with other type of information. Since a homepage could be represented by a logical page group, it is necessary to consider not only the content of the entry pages but also the information in the surrounding pages based on page group structures (local link structures).

We proposed a method to build a high-quality homepage collection which uses Support Vector Machine (SVM) by utilizing the features considering page group structures and used researchers' homepages as an example category. We expect the method to be used not only to the category of researchers' homepages but also to other categories of homepages. Since the number of web pages is very large, it is natural that the computational cost of feature extraction is very high. We therefore split the process into two steps: an initial high-recall rough filtering that efficiently narrowing down the amount of candidate pages, and then a high-recall high-precision classification of the candidate target pages output from the rough filtering. Both processes take into consideration page group structures and utilizing useful information based on the web structures, so that we can achieve a high classification performance, especially in terms of recall, for meeting the quality requirement for the collection. Besides, since high classification performance usually means high processing cost, the processing cost must be taken into consideration too.

One of our unique contributions is that we pointed out the importance of assuring the quality of the web page collection and proposed a framework for assuring the quality. Another is that we introduced an idea of local page group models (PGMs) and demonstrated its effective uses for filtering and classifying web pages.

1. We presented a realistic framework for building a high-quality web page collection in two steps: rough filtering followed by accurate classification.

2. For the rough filtering, we proposed page group models (PGMs) and introduced two key techniques to modify the PGMs so that a very high recall can be assured with little noise.

- Use the number of out-linked pages in the same and lower directories as the threshold to propagate the keyword lists;
- Introduce two types of keyword lists and to propagate only the organization-related keyword lists from the upper directories.

The noises introduced by using the PGMs are reduced to almost the same level as that of using single page model(SPM). Besides, the ability to find a considerable number of potential target pages by using PGMs but can not by using SPM further shows the effectiveness of the key techniques.

3. In the accurate classification, the key techniques are the feature subset concatenation and the composition of the three-way classifier.

   - Exploit features from the surrounding pages and concatenate the features independently instead to merge them for improving the classification performance;
   - Use a recall-assured classifier and a precision-assured classifier in combination as a three-way classifier for reducing the amount of pages requiring manual assessment under the given quality constraints.

   The performances of the classifiers are improved significantly by using the proposed features and the number of pages that require manual assessment for assuring the required quality is reduced notably by using the three-way classifier.

## 1.2 Position of the Research Work

### 1.2.1 Research-related Information and Systems

Almost every kind of information we can image is provided on the web, most of it can be found by search engines [51] [52], and some web services search mainly for research-related information.

Research-related information can be classified into the following four categories:

1. Metadata on academic publications, including both books and papers, such as that provided by DBLP[6], PubMed[7], CiteSeer, and Web of Knowledge.

   The information about books and papers that is provided on the web is well structured and often available in databases. It is indexed by various characteristics, and web services enable it to be searched with reference to article title, author name(s), conference, journal, series, publisher, subject, etc. Even though some search services, like CiteSeer and DBLP, stress both the precision and recall, they do not guarantee them.

2. Metadata on researchers, such as that provided by ReaD and Web of Knowledge.

   The information is provided separately or with other related information and always has a well-defined structure. ReaD, for instance, provides information about the research topics, projects, publications, and affiliations of several hundred thousand researchers in Japan. Its information is collected by questionnaires, however, a considerable part of the information is not up-to-date. It is a independent information service having no links with academic publication information. Web of Knowledge, for another instance, provides information on authors together with information on papers.

3. Researchers' homepages, such as those provided by ReaD and "The Comprehensive Compendium of Plankton Researcher Homepages "[8].

   Unlike the researchers' information provided by databases, the information on researchers' homepages is not structured. Researchers' homepages are also very difficult to provide because they are hard to collect. For example, The Comprehensive Compendium of Plankton Researcher Homepages contains only several hundred homepages and most of them are submitted by the researchers themselves. Researchers can provide their homepages when they fill

[6]DBLP. http://www.informatik.uni-trier.de/ ley/db/index.html.
[7]PubMed. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pmc.
[8]http://www.biosci.ohiou.edu/faculty/currie/ocean/plankton-res.htm.

out the ReaD questionnaire, but a lot of them fail to provide their homepage URLs.

4. All the other kinds of research-related information: project information, conference information, funding information, research tools, etc.

Although some web services provide researchers' homepages, the numbers of pages they provide is very small because there is no way to collect them efficiently. Therefore the goal of the work presented in this dissertation was to investigate a method for efficiently collecting researchers' homepages from the Web.

If the target collection of researchers' homepages could be obtained efficiently, it could be used to provide up-to-date and more detailed information about more researchers. It would also be useful to researcher-related information services, as a source of information for compiling and maintaining researchers' databases, as a resource for domain-specific search engines, etc.

The popular web service systems usually provide these kinds of information in some combined ways. In CiteSeer, for example, a paper's page can link to the pages of other papers that have a similar topic, that are cited by the paper, or that cite the paper. In Web of Knowledge a paper's page not only has links to the pages of related papers just as in CiteSeer but also has links to pages that include the author's and the co-author's information, and those information pages have links to the pages of the author's papers.

When the researchers' homepages are available, not only can links to the authors' homepages be added to the author's paper's pages but also researchers information more timely than that obtained by way of ReaD's questionnaires can be obtained from up-to-date researchers' homepages.

### 1.2.2 Quality-guaranteed Applications

Many others have explored ways provide the classification performance needed for assembling a high-quality collection, but no one has yet assure the quality of the collection.

There has been a lot of research on quality control for conventional databases. The quality of databases on research paper and company finance, for instance, is improved by using methods for collecting high-quality data sets.

Almost all the applications related to web information have only been best-effort types, however, in part because assuring the quality of a web page collection is labor intensive and time consuming. Nevertheless, there are many potential applications that require data collections with a guaranteed high level of quality.

We are therefore trying to investigate a method for building a high-quality collection of researchers' homepages that can be used for extending quality-guaranteed information service, such as CiNii, Web of Knowledge, and Scopus[9]. Such a collection would be used not only as a source of link targets but also as reference data for record linkage [53] the authors' research papers, citations, and/or research project reports, etc.

The method for building the homepage collection should thus be able to assure the quality—both the completeness (high recall) and the accuracy (high precision)—that are required by the nature of the applications. Even with state-of-the-art technology, however, the required quality can in most cases not be assured with computer processing alone. We therefore investigate a method with which all the web pages are accurately classified into three classes: assured positive, assured negative, and uncertain. The assured positive category corresponds to the required precision and the assured positive and uncertain categories correspond to the required recall. The pages classified as uncertain need to be assessed manually, so their number must be minimized.

### 1.2.3 Collecting Web Pages

There are many ways to collect web pages on a given topic/category.

Focused crawling gathers web pages on a topic of interests by using information available in the link source pages, such as URLs, global and/or local anchor texts, etc. It tries to collect as many relevant pages as possible in a limited page amount,

[9]Scopus. http://www.info.scopus.com/.

emphasizing the accuracy (high precision) but not the completeness (high recall), and is consequently evaluated by the efficiency of the crawling.

Much research on web information retrieval and web page classification has been focused on improving the performance. Domain-specific search engines (e.g., DBLP and CiteSeer), build their collections by using information retrieval, text classification, and focused crawling, but they are only best-effort services because the techniques they use do not support quality assurance.

The first step of our method, rough filtering, efficiently gathers candidate pages in a specific category from the web and is thus similar to focused crawling. However, although it also stresses efficiency, priority is given to high recall with some constraint (e.g., 98% or 99% recall) in order to assure the quality. The second step, accurate classification, is also a kind of web page classification, but it provides means for assuring both the recall and precision. The collection built using our method therefore can be used to provide quality-guaranteed services.

## 1.3 Scope of the Research

The goal of our research was to study a method for building a high-quality homepage collection from the web and it focused on collecting and classifying web pages. To make a web page collection ready to use in real applications, further problems (e.g., creating the metadata, determining the extent of web page groups, identifying and disambiguating entities) must be solved. And if the collection is to be a target for links to the authors of a paper, the authors need to be further identified in the collection.

These problems are as important and difficult as collecting the web pages is, but they might be easier to solve after a high-quality homepage collection is obtained. We therefore focused on collecting web pages.

## 1.4 Organization of This Dissertation

This Ph.D. dissertation presents the method for building a high-quality homepage collection from the web efficiently by considering the page group structures. We investigated various categories of homepages but mainly researchers' homepages.

This dissertation consists of seven chapters. Chapter 1 gives the introduction, and Chapter 2 presents the related work. Chapter 3 describes the objectives, the overall performance goal of the investigated system, and the scheme of the system. Chapters 4 and 5 discuss the two-step-processing method, and Chapter 6 further discusses the ways used to reduce the processing cost. Chapter 7 concludes this dissertation and discusses future work.

Chapter 1 briefly introduces the background, the objectives, the approach, and the originalities of the research. Chapter 2 reviews the related work and compares it with our work.

Chapter 3, taking into account the enormous size of World Wide Web, introduces our two-step-processing method comprising rough filtering and accurate classification. The former is for efficiently narrowing down the amount of candidate pages while maintaining the required high recall, and the latter is for accurately classifying the candidate pages into three classes—assured positive, assured negative, and uncertain—while assuring the required recall and precision.

Chapter 4 presents the detailed configuration and the experiment evaluation of the rough filtering. The rough filtering is a method for gathering researchers' homepages (or entry pages) by applying our original simple and effective page group models. These models enable us to exploit the mutual relations between the structure and the content of a page group and thereby narrow down the amount of candidate pages while assuring a very high recall. First, 12 property-based keyword lists that correspond to researchers' common properties are created and are assigned as either organization-related or non-organization-related. Next, taking into consideration the structure in an individual logical page group, four page group models (PGMs) are introduced: PGM_Od, PGM_Ou, PGM_I, and PGM_U. Based on the PGMs, the keywords are propagated to a potential entry page from its surrounding pages, composing a virtual entry page. Finally, the virtual entry pages that scored at least

a threshold value are selected. Since the application of PGMs generally causes a lot of noises, we introduced four modified PGMs with two original techniques. The four modified PGMs are used in combination in order to utilize as many informative keywords as possible from the surrounding pages. The effectiveness of the method is shown by comparing it to a single-page-based method through experiments using a 100GB web data set and a manually created sample data set.

Chapter 5 presents the detailed configuration and experiment evaluation of the accurate classification method. Using two types of component classifiers (a recall-assured classifier and a precision-assured classifier) in combination, we construct a three-way classifier whose input is the candidate pages output by the rough filtering and whose output is three classes: assured positive, assured negative, and uncertain.

We first devise a feature set for building the high-performance component classifiers using Support Vector Machine (SVM). We use textual features obtained from each page and its surrounding pages. After the surrounding pages are grouped according to connection types (in-link, out-link, and directory entry) and relative URL hierarchy (same, upper, or lower in the directory hierarchy), an independent feature subset is generated from each group. Feature subsets are further concatenated conceptually to compose the feature set of a classifier. We use two types of textual features, plain-text-based and tagged-text-based. The classifier using only the plain-text-based features in each page is used as the baseline. Various combinations of the feature subsets were tested in the experiment using manually prepared sample data, and the classifiers were tuned by two methods: an *offset*-based one and a *c-j-option*-based one.

To evaluate the performances of the classifiers with the above mentioned feature sets in more general cases, we experimented with our method on the Web->KB data set, a test collection commonly used for the web page classification task. By tuning well performing classifiers independently, we then built a recall-assured classifier and a precision-assured classifier and compose the three-way classifier by using them in combination. We estimated the numbers of the pages to be manually assessed for the different levels of required precision and recall by using the output pages obtained by roughly filtering a 100GB data set. We also evaluated the effectiveness of the

classifiers by analyzing the classification result examples.

Chapter 6 describes the cascaded structure of the recall-assured classifiers, used in combination with the rough filtering, we use to reduce the computer processing cost. An estimation of the amount of the pages requiring feature extraction in the accurate classification confirms that the cascaded structure can reduce the computer processing cost evidently.

Chapter 7, based on the experiment results and related considerations, concludes that the proposed two-step-processing method performs as required. This chapter also briefly summarizes our contributions and indicates directions for our future work.

# Chapter 2

# Related Work

Numerous web pages are available online and the World Wide Web is becoming a more fertile area for data mining research.

Web mining is a technology for finding information in web pages and extracting it [54] [55] [56]. The lack of data structure in web pages makes the automated discovery of targeted or unexpected information a challenging task. Web mining can be classified into three catogories [57]: Web usage mining, or the discovery of user access patterns from usage logs; Web structure mining, or the discovery of useful knowledge from the structure of hyperlinks; and Web content mining, or the extraction of information from web page contents.

In this research we try to collect all the probable researchers' homepages with as few irrelevant pages as possible by exploiting the relations between the structures and contents by taking into consideration page group structure among a huge amount of the web pages. Our collection method, which can be regarded as a technique related to both web structure mining and web content mining, provides a high-quality homepage collection by assuring both a specific high recall and a specific high precision.

Since, as mentioned in Chapter 1, our final goal is to collect all the probable researchers' homepages with as low a processing cost as possible from the huge amount of web pages, we investigated the feasibility of first roughly filtering all the pages for narrowing down the amount of candidate pages and then accurately classifying those pages.

As will be described in Chapter 4, the rough filtering could be regarded as a web page search method as well as a web page classification method, both using property-based keywords and considering of page group structures. The accurate classification could be regarded as a web page classification method by exploiting the content information in the surrounding pages while taking into account their location relative to the target page.

## 2.1 Web Information Retrieval

Our research aims at collecting researchers' homepages and there is much related work on the use of web information retrieval methods directly aiming at homepage finding. There are many related studies and those mentioned below are just examples.

Xi et al. proposed a method for predicting the correct homepage in response to a user's homepage finding query by taking into consideration the most likely URL of the homepage [33]. Experimental results show that 84% out of 145 testing queries had the correct homepage returned within the top 10 pages, while only 59% of the testing queries had their correct answers returned within top 10 hits without using the method. Harada et al., focusing on finding authoritative people, presented a method to extract proper names from the web pages retrieved in response to a topic query by utilizing the number of web servers containing a name instead of the number of web pages [58]. Shakes et al. presented the Dynamic Reference Sifting method, which for some specific page categories tries to provide both maximally comprehensive coverage and a highly accurate response in real time [59]. Their method uses Homepage Finder Ahoy![1] to filter out the most likely one or two references that point to the person's homepage. If there is no likely candidate, Ahoy! can use knowledge of homepage placement conventions to "guess" the URL for the desired homepage.

Artiles et al. described the strategies for searching for people on the World Wide Web by resolving person names' ambiguity and locating relevant information

[1] Ahoy!. http://www.cs.washington.edu/research/ahoy.

characterizing every individual with the same name [60]. Doring used a communication studies framework for finding personal homepages [14]. It regards personal homepages as media products with specific production processes, product characteristics, and reception processes. Kang and Kim proposed a method for getting better retrieval results for the homepages of persons by using content information, link information, and URL information [61].

The first three of these six methods use both the query terms and the URL. The fourth one uses the person name as the query and uses link information to distinguish the pages of different people with the same name. The fifth one uses page characteristics related to contents, representations, URLs, etc. The last one uses, in addition to the content information, not only URL information but also link information.

Chakrabarti et al. suggested a focused crawling approach for building high-quality collections of web documents on specific topics [62]. To be able to answer all possible ad-hoc queries, a focused crawler dose not collect and index all accessible web documents but instead uses a classifier to analyze its crawl boundary and find the links that are likely to be most relevant for the crawl and then uses a distiller in order to avoid irrelevant regions of the web. These investigators proposed a method for collecting web documents on specific topics by utilizing the relationship of the query with the neighbor linked pages.

Oyama et al. described a method of searching for particular categories of web pages by adding domain-specific keywords called "keyword spices" to the input query and forwarding them to a general-purpose search engine [63]. A practical learning algorithm implemented by only a small number of Boolean expressions is used for extracting powerful but comprehensive keyword spices. The effectiveness of keyword spices was shown by the results of experiments with the keyword spices for the domains of recipes, restaurants, and used cars. Mori et al. proposed a keyword extraction method to extract Friend of a Friend (FOAF) metadata from the web [9] [64]. The method is based on the co-occurrence information of words and it extracts relevant keywords depending on the context of a person. The experimental results show that extracted keywords are potentially useful for extracting FOAF

metadata from the Web. In the method of Oyama et al. keyword spices are used to reduce the number of irrelevant pages on rather specific topics, while in the method of Mori et al. keywords are extracted as the metadata of persons and their relationships.

Zhang et al. proposed a method that benchmarks three methods—TFIDF, KEA, and Keyterm—used to extract key phrases from all the plain text of web pages and from only the narrative text [65]. Their evaluation of the ranking data shows that key phrases extracted only from the narrative text are significantly better than those obtained from all the plain text in web pages.

Most of the research on methods for web information retrieval has tried to take advantage of various kinds of web-based information.

Matsuda and Fukushima introduced a method for task-oriented web information retrieval by classifying documents according to various page characteristics (content, out-link, URL, and so on) and succeeded in reducing the retrieved number of task-irrelevant pages [50]. Kleinberg investigated a method that reduces the number of pages retrieved in response to broad search topics by finding only "authoritative" information sources of information about the topics [16]. He proposed and tested an algorithmic formulation of the notion of authority, that was based on the relations between a set of relevant authoritative pages and the set of "hub pages" that join them together in the link structure.

In addition, Khan and Locatis investigated the relations between search performance, hyper text format, and the number of hyperlinks [66]. Both link density (number of links per display) and display format (in paragraphs or lists) were found to have significant effects on search performance. Low link densities displayed in list format produced the best overall results in terms of search accuracy, search time, number of links explored, and search task prioritization. The out-link number was used for focusing on the target page easily. Eater et al. similarly discussed the classification of the complete web site [27]. They treated a web site as a large HTML-document with a lot of subtrees and improved classification accuracy by using a powerful pruning method to reduce the number of web pages considered.

All four of these investigations found that the performance of web page retrieval

could be improved by taking into account the global link structure, the latter two found it could be further improved by also taking into account the number of links.

On the other hand, Liu et al. proposed a web data cleansing method with non-content features including but not limited to link analysis features [67]. The key source pages are automatically selected by taking into consideration the in-site out-links because that key resource pages should have enough in-site out-links to connect to other pages and enough in-site out-link anchor text to give a brief view of these pages. As a result, the 44.3% of all pages that are selected as the high-quality collection contain more than 98% of the links and cover about 90% of the key information. This method uses the local link structure within the same site without using the content-related information. Li et al. presented an algorithm for efficiently retrieving information units that can perform progressive query processing over a web index by considering both semantic similarity and link structures [68]. The work tried to utilize some measures obtained by analysis of the structures among the web pages for improving the information retrieval performance. Both the above two works exploited local link structure and showed the effectiveness of the method on improving the performance.

Masada et al. proposed a method for improving web search performance with a Vector Space Model exploiting hyperlink information by means of a new web page clustering algorithm [69]. The clustering results are used for modifying entries of document vectors. Their experiment results show that the average precision can be improved by more than 10%. Tajima et al. developed a technique for discovering and retrieving the logical information units in web data [49]. Regarding the given conjunction queries, they try to approximate information units including all the given keywords in the following three steps: (1) distinguish standard route links from the others, (2) find minimal subgraphs connected via those links and including all the keywords, and (3) compute for each subgraph a score based on the locality of the keywords within it in order to determine whether it really is a logical information unit relevant to the query. This technique is effective when all the keywords in the queries are known beforehand. The work of Masada et al. as well as that of Tajima et al. exploits the subgraph of the web pages, and that of Tajima et al. further

utilizes the information related to local link structure to confirm the logical unit of a web page.

All the work referred to in this section, except the work on the extraction of keywords, exploits structural information (i.e., global/local links, anchor text and URL hierarchy) in some way with or without using page content information.

## 2.2 Web Page Classification

Web page classification has numerous applications in the hypertext and semi-structured data domains, and the methods for web page classification and the like generally try to exploit various types of web-related information sources for feature extraction, such as textual content, html tag, URL, and the directory/hyperlink structure.

The method using only the features based on the textual content in the target page alone is the simplest one [70] and is always used as a baseline for performance comparisons.

Bekkerman et al. generated a compact and efficient representation of documents by using, instead of the simple bag-of-words representation, the distributional word clusters computed using the Information Bottleneck method [71]. Combined with Support Vector Machine (SVM), this method yielded high performance in text categorization with three known data sets for web data categorization.

To exploit features of html tags and anchor texts, Sun et al. proposed the use of Support Vector Machine (SVM) classifiers to classify web pages using both their text and context feature sets, such as hyper links and html tags [72]. The results of experiments using the Web->KB data set showed this method works very well. Sun et al. applied features such as plain texts, hyperlinks, and anchor texts to GECKO (Generalized Eigenvalue based Composite Kernel Optimization), an optimized linear combination of kernels [73].

Shih and Karger proposed some new features using URLs and table layouts for web classification tasks, such as content recommendation and ad blocking [32]. Instead of looking at the textual content in the pages, they look at each link's URL and take into consideration the visual placement of those links on a referring

page. A fast webpage classification method using only URL features was proposed by Kan and Thi [74]. By segmenting the URL into meaningful chunks and adding component, sequential, and orthographic features to model the salient pattern, the approach is faster than the typical web page classification.

Calado et al. reported that global link information, such as co-citation, can be used to obtain a high-performance of web classification [24], and Wang and Kitsuregawa showed that in-link reinforcement and anchor windows can improve the quality of web page clustering by utilizing the global link structure efficiently [23]. Both of these groups of researchers have shown the effectiveness of classification using global information.

Chakrabarti et al. proposed a method for enhanced hypertext categorization by using hyperlinks [75]. Although using link information is noisy, using information about the links in a small neighborhood around the document can reduce number of the classification errors by up to 70% from the textual-content-based classifiers. Sun et al. reported that a classifier using html tags (title) and hyperlinks (anchor text) as web page features, in addition to textual-content-based features with SVM, resulted in good classification performance on the Web->KB data set [72]. Glover et al. also developed a method for describing and classifying web pages by using web structure [26]. Their method analyzes the relative utility of document text and the text near the citation in citing documents. Regarding words and phrases in the citing documents according to the expected entropy loss enables the web pages to be clustered accurately even with very few positive examples. Chau proposed a machine-learning-based approach that combines the features related to web content and web structure for the classification of a large collection with only a small number of training examples [76]. The features considering the content in the page itself and in the neighbor pages and the page's link information are used in the classifier. The abovementioned works thus use for classification not only textual-content-based features but also features on information about global links.

Sun and Lim proposed an iterative web unit mining method for finding and classifying web units of web pages [48]. In consideration of the page structure, sub-units are merged and then the key pages of the units are classified. The page content is

used only for classifying individual pages and is indirectly combined with structure-related information. This method uses features related to local link structures as well as features on the contents of target pages.

All the methods referred to in this section, except the last one, are used to capture the features that are characteristic to the target pages and are effective for selecting highly probable pages. The last one, in contrast, is used to collect dispersed information and is effective for gathering potential pages comprehensively.

## 2.3 Conclusion

To make a high-performance classifier that can be used for building a web page collection while assuring quality required for its application—that is, a collection satisfying the given recall and precision requirements—we focused not only on obtaining high-performance classification by finding an effective feature set but also on minimizing the manual assessment needed to assure both the recall and precision of the collection.

Although others have shown the effectiveness of utilizing various kinds of information (URL, in-/out-link, image, etc.), we tried to find effective features of textual-based contents. That is, we tried to find a better way to use textual features for the classification. We therefore proposed a classification method that uses plain-text-based and tagged-text-based features which can be easily combined with other methods.

When the homepages are presented in logical page groups, the information both in the target pages and surrounding pages should be taken into account when homepages are being classified. We therefore decided to exploit the features of various local surrounding pages (in-linked pages, out-linked pages, and directory entry pages in the directory path to the site root) in combination with the relative directory level of the pages (in the same directory, in the upper directories in the URL path, or in the lower directories of the directory subtree). Since utilizing the information in dispersed pages tends to increase noises, we create the features on surrounding pages concatenated but not merged and use them in the classification all together

so that the contexts corresponding to the relative location can be well represented.

Although there are many web page classification methods, they all focus on achieving the high performance with regard to general measures, such as F-measure, and mainly on precision but little on recall. Furthermore, they provide no means to assure the quality. Since it is impossible to achieve both high precision and high recall at the same time, we decided to use a recall-assured classifier and a precision-assured classifier in combination in order to assure the recall and precision required for assuring the quality of the target homepage collection. Because satisfactorily high performance cannot be obtained automatically by using these two kinds of classifiers, their performance must be improved in order to reduce the amount of manual assessment required to assure a given quality level.

# Chapter 3

# Scheme of the Method

## 3.1 Objectives

Since a homepage can be not only a single page but also an entry page of a logical page group, we needed to investigate a system for gathering both kinds of homepages. As it is impossible to determine the extent of a logical page group in general cases, we did not try to identify the logical page groups, but simply to gather their entry pages.

The quality of the homepage collection should be guaranteed because the collection is expected to be used as a web resource for quality-guaranteed applications. In the present research, we regarded the requirements for the quality of the target collection to be specified in terms of recall and precision.

The processing cost naturally becomes higher when we try to obtain higher quality, so we needed to take into account the overall processing cost of the system fulfilling the quality requirement. A collection with the required quality should be built with as low a processing cost as possible.

Since it is difficult to obtain the required high performance even when using state-of-the-art classification techniques, manual assessment is inevitable. We therefore had the following three objectives:

1. To implement high-performance classification techniques;

2. To reduce the manual assessment cost;

3. To reduce the computer processing cost.

And although we were trying to develop a method for building a high-quality collection of researchers' homepages in Japanese, we wanted our method to be a general one that can also be applicable to build collections of homepages of other categories in other language.

## 3.2 Performance Target

The quality requirement of a homepage collection depends on the practical application of the collection and can be specified by various combinations of precision and recall, such as 99.5% precision and 98% recall, 99% precision and 95% recall, or 98% precision and 90% recall.

As explained in Chapter 1, we wanted to investigate a method for obtaining a high-quality collection of researchers' homepages that can be used for extending a quality-guaranteed information service like CiNii. Taking these applications into consideration, we set the overall target performance of the investigated system at 95% recall and 99% precision.

Using the manually created positive and negative samples that will be described later (Chapter 4), we did a preliminary experiment in which researchers' homepages were classified using SVM$^{light}$[1] with features based only on the content words in individual pages. This resulted in a recall much lower than the precision: 59.60% recall and 90.15% precision.

Since the amount of web pages is enormous and classifying all the pages is computationally very expensive, we need to reduce the number of pages need to classified to a feasible level. And because the manual assessment needed for assuring the quality of the target collection is much more expensive than the computer processing, we also need to reduce it as much as possible.

[1] SVM$^{light}$ Support Vector Machine. http://svmlight.joachims.org/.

## 3.3 System Structure

The need to assure the quality of the target homepage collection at the lowest possible processing cost make the efficiency of the system configuration very important.

The target collection will contain the homepages either in single pages or in logical page groups, and some research shows that it is generally effective to collect them by using the features exploiting link structure, directory structure, document tag structure, and document semantic structure. Processing these features for all the web pages, however, is too costly. We therefore split the process into two steps and structured the system as shown in Figure 3.1.

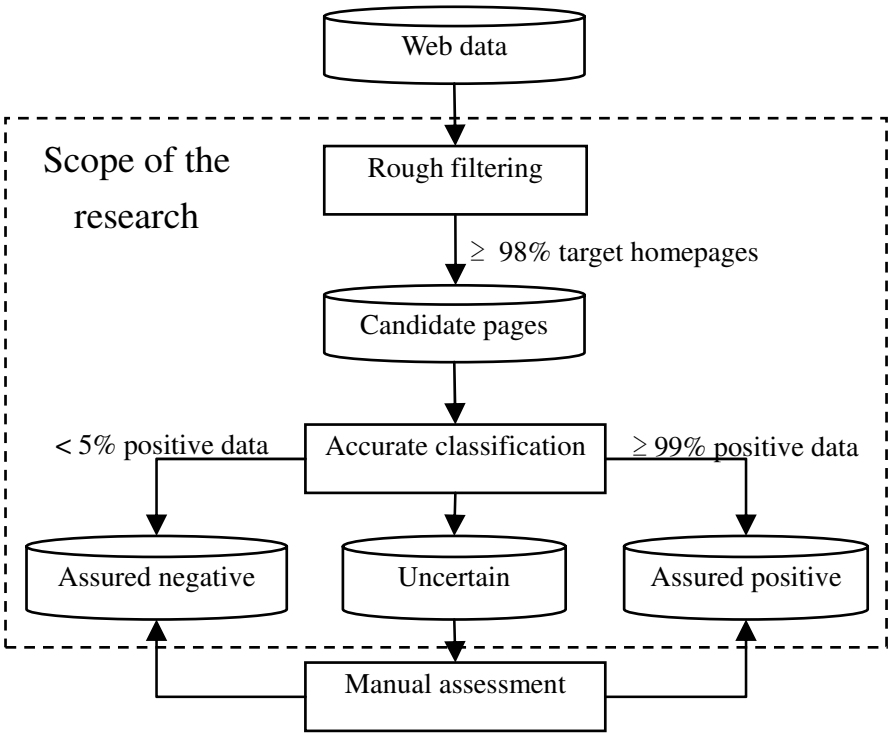


Figure 3.1: System structure.

- **Rough Filtering**

  The rough filtering is for efficiently narrowing down the number of candidate pages needed to assure a very high recall. The input is all the web pages and the output is the candidate pages satisfying the specified level of recall.

To achieve the overall performance goal given in the previous chapter, we set the performance required for the rough filtering as at least 98% and desirably 99%. Precision does not matter so much but a smaller amount of output pages is preferable to a larger amount.

- **Accurate Classification**

  The accurate classification assigns the candidate pages output from the rough filtering into three classes: assured positive, assured negative, and uncertain. The recall should be at least 95% and the precision should be at least 99%.

  Since even with state-of-the-art classification techniques it is impossible to assure the required quality of the target collections solely by means of automatic processing, human involvement in the classification process is indispensable.

  The computer processing cost for the accurate classification can be higher than that for the rough filtering but the number of pages that need manual assessment needs to be made as small as possible.

Note that the rough filtering filters out the definitely irrelevant pages in order to efficiently narrowing down the number of candidate pages that need to be further classified accurately. It is an essential process when the number of web pages being processed is very large because applying a huge amount of web pages to a classifier is too expensive. When the number of web pages being processed is not very large, however, the rough filtering can be omitted and only the accurate classification need to be performed.

## 3.4 Basic Concepts

### 3.4.1 Researcher's Homepage

Although researchers' homepages use various styles of presentation, they all contain several basic elements. We define a researcher's homepage as a web page whose main subject is the researcher and that includes the researcher's personal attributes (such as the name, affiliation, address, major, and academic societies) and research

activities (such as research subjects and publications). This information may be provided either in a single page or in a logical page group, in which case the entry page is regarded as the homepage. Some personal attributes and professional activities might not be specified, but a researcher's homepage must provide enough information to identify the researcher and to outline his/her research activity. A researcher's homepage is usually created by the researcher or the organization to which the researcher belongs. Our current method filters only homepages written only in Japanese as an example and classifies homepages written in Japanese or in English.

### 3.4.2 Logical Page Group

A logical page group is a group of web pages that consists of an entry page and one or more component pages, none of which alone contains sufficient information on the topic. Only the entry page together with the component pages contain sufficient information. The component pages are not necessarily directly or indirectly accessible from the entry page. For example, if an entry page does not include the organization's name but the site top page includes it, then the site top page is considered to be a component page even if the entry page does not contain a link either to the site top page or to a page that contains a link to the site top page. In the current work, a component page has to be either an in-linked page, an out-linked page, or a directory entry page in the directories included in the URL path of the entry page and has to be in the same site.

## 3.5 Data

For the experiments on both the rough filtering and the accurate classification, we used a 100-GB corpus of web document data, NW100G-01[2], which was gathered from the '.jp' domain for WEB Tasks[3] [77] [78] at the Third and Fourth NTCIR

[2]NW100G-01 is available for research purposes from the National Institute of Informatics. See http://research.nii.ac.jp/ntcir/permission/.

[3]NTCIR WEB Task (NTCIR-WEB), http://research.nii.ac.jp/ntcweb/.

Workshops [79] [80]. It contains 11,038,720 web pages. We used the link list attached to the document data and the full-text index of the document data generated by "Namazu"[4]. Note that when the keyword lists described later are fixed, the web crawling process can without loss of efficiency use a string-matching algorithm like Aho-Corasick's[5] instead of the full-text index.

To evaluate the effectiveness of the rough filtering we also used a 1.36-TB ($1.5 \times 10^{12}$ byte) corpus of web document data, NW1000G-04, which was created for the WEB Task at the Fifth NTCIR Workshop [81] and contains 95,870,352 web pages.

Both of these data sets were also used for evaluating the processing cost of the total system.

[4]Namazu: a Full-Text Search Engine. http://www.namazu.org/index.html.en.

[5]Aho-Corasick Algorithm. http://en.wikipedia.org/wiki/Aho-Corasick_algorithm.

# Chapter 4

# Rough Filtering

## 4.1 Introduction

This chapter describes a rough filtering method for gathering researchers' homepages with a very high recall by applying page group models combining the structure and content of the pages in a page group. As described in Chapter 3, it is the first processing step in our homepage collection method. It is efficient enough to process the huge amount of web pages and can gather probable candidate pages with a very high recall while the output can be reduced to a reasonable low amount.

The effectiveness of the rough filtering based on the page group models (PGMs) comparing with that of a single-page-based method was shown by the results of experiments with various parameters using the 100GB and 1.36TB data sets as well as a manually created sample data set described in Section 4.4.

The rest of this chapter is organized as follows. Related work is introduced in Section 4.2, and the structure of the rough filtering is described in Section 4.3. The manually created sample data is described in Section 4.4, and the property-based keyword lists and page group models with the considered parameters are discussed in Sections 4.5 and 4.6. Section 4.7 describes the experiment results, Section 4.8 discusses them, and Section 4.9 concludes the chapter by briefly summarizing it.

## 4.2 Related Work

The rough filtering efficiently narrows down the number of candidate pages while missing as few target homepages as possible. Taking into consideration the web page group structures, we use a set of property-based keywords to match the contents not only of each page itself but also of its component pages. The rough filtering could be considered as a web page search method as well as a web page classification method.

Oyama et al. described a method for searching the web pages of particular categories by adding domain-specific keywords called "keyword spices" to the input query and forwarding them to a general-purpose search engine [63]. Our use of property-based keywords is in a sense similar to their use of keyword spices, but their purpose is to reduce the number of irrelevant pages on specific topics, whereas our purpose is to gather all the possible pages of a given category.

Matsuda and Fukushima introduced a method for task-oriented web information retrieval utilizing document classification according to various page characteristics (content, out-link, URL, and so on) and succeeded in reducing the number of task-irrelevant pages [50], but all of the characteristics they use are extracted from individual pages and no characteristics of page group structure is utilized.

Li et al. presented an algorithm to efficiently retrieve information units that can perform progressive query processing by considering both semantic similarity and local link structures [68]. Masada et al. proposed a method for improving web search performance with a Vector Space Model exploiting the contents of the local out-linked pages instead of using the contents of global linked pages by means of a new web page clustering algorithm [69]. The algorithm of Li et al. and the method of Masada et al. both utilize some measures obtained by analyzing the local link structures of the web pages for reducing the number of irrelevant pages in the search results. Our method, in contrast, uses the directory level related to the page group structure to collect useful keywords from the surrounding pages for gathering candidate pages comprehensively while keeping the number of irrelevant pages small.

Calado et al. reported that global link information, such as co-citation, can be

used for high-performance classification [24], and Wang and Kitsurekawa showed that in-link reinforcement and anchor windows can improve the quality of web page clustering by utilizing the global link structure efficiently [23]. However, the effectiveness of global information for achieving high recall is limited since many noises are introduced in the same time.

Sun and Lim proposed an iterative web unit mining method for finding and classifying web units of web pages [48], and Tajima et al. developed a technique for discovering and retrieving the logical information units in web data [49]. In the method of Sun and Lim, however, the individual page content is used for classifying pages and is indirectly combined with structure-related information; and in the technique Tajima et al. investigated the minimal subgraphs are targeted for answering the queries. Those methods are therefore of only a limited effectiveness for the improvement of the recall.

Even though the abovementioned methods exploit structural information (i.e., global/local link, anchor text, and URL hierarchy) in some way with or without page content information, none of them use page group structure to exploit information presented on several related pages. Although others might have tried to use an idea like ours in web page searching or other tasks, we know of no one who has made the idea work properly in general cases because of the difficulty in excluding irrelevant information or noises.

Nevertheless, our method does use the idea and works to a certain degree. The reason is that, as will be shown later, our method exploits the mutual relations between the page content and the relative page location in generating virtual pages by propagating keywords. It works because of the original techniques we have introduced for modifying the simple PGMs.

## 4.3 Structure of the Rough Filtering

The method we proposed uses property-based keyword lists and several kinds of page group models. Figure 4.1 illustrates its conceptual structure.

Each web page is first mapped to a **document vector** consisting of binary

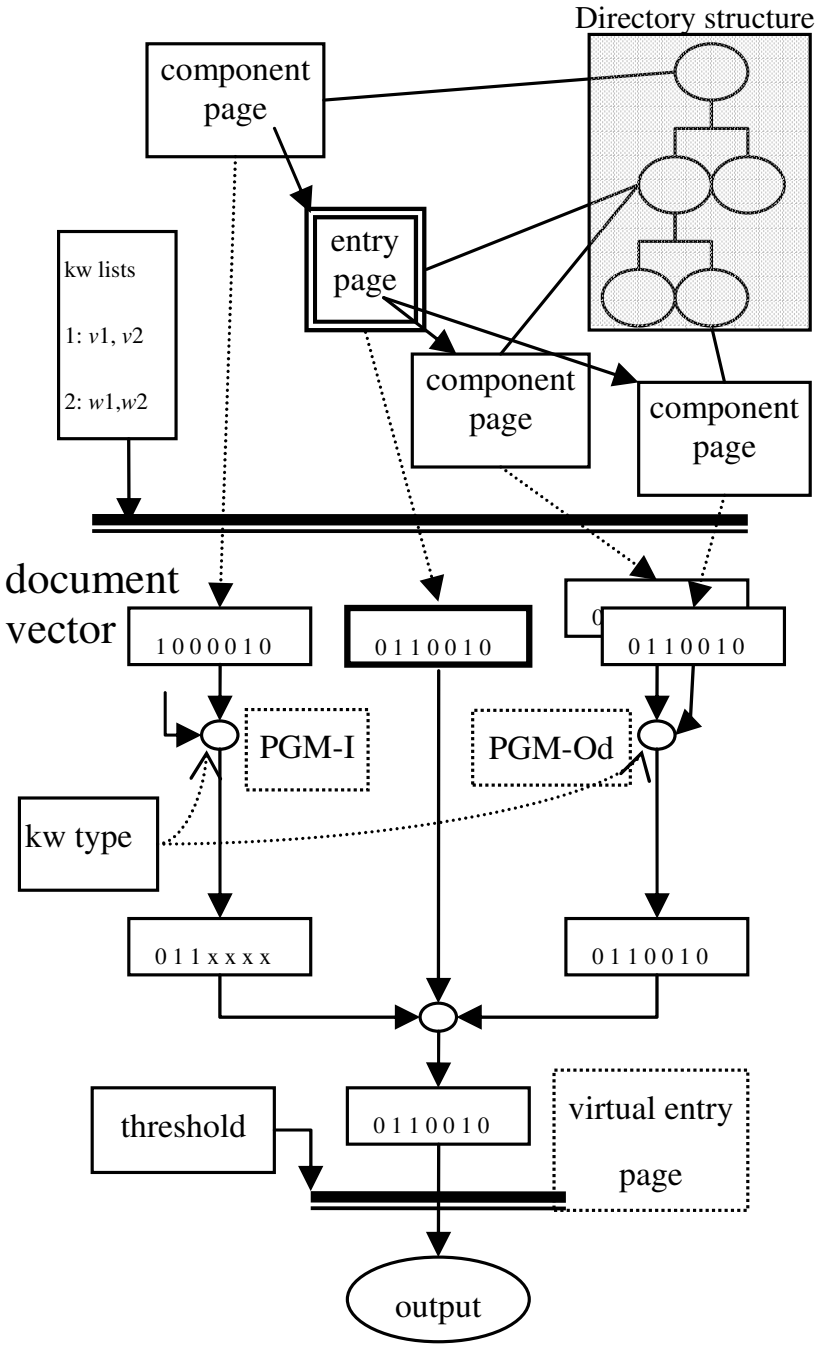


Figure 4.1: Structure of the rough filtering.

values, each of which corresponds to a keyword list and represents whether or not any of the keywords in the keyword list are in the web page.

Then the document vectors for each of the page group models are merged by making the logical sum of each vector element. In this process, only the elements corresponding to the suitable type of keyword lists for the page group models are considered (at the output from PGM-I each of the ignored elements is indicated with an 'x'). They are further merged to the page's document vector in the same way to compose a final document vector. A conceptual document represented here by the final document vector is called a **virtual entry page**, and the process used to merge the document vectors of the individual pages to that of a virtual entry page is called **keyword propagation**.

The **scores** of virtual entry pages are obtained by counting the number of 1's in the document vector, and the pages with scores equal to or greater than a **threshold score** are output. The threshold score will be set to as high as possible by considering the evaluation results so that the recall satisfies the recall requirement, e.g., 99%.

It should be noted that we do not use popular techniques like calculating a TF-IDF score or using probabilistic models because term frequency makes no sense for property-based keywords (only their presence or absence does) and property-based keywords are essentially popular terms and their specificities are of no use.

## 4.4 Sample Data

To prepare sample data from the NW100G-01 document set, we first collected 113,380 pages containing some typical Japanese family names and randomly selected 10% of them (here we call this set of 11,338 pages the **Jname data**). Each of the pages was then manually assessed (by me) according to its content and, if necessary, the contents of its in/out-linked pages. The Jname data was thus partitioned into 426 positive samples and 10,912 negative samples that were used for the experiments with the rough filtering.

It should be noted that because the rough filtering stresses recall, the sample data must be prepared with sufficient care to not introduce biases in any aspects. If

we use precision-assured methods to collect possible pages that are to be manually assessed, more "easy" pages and fewer "difficult" pages would be collected. The final recall would thus be overestimated. Since we could find no other bias-free way to efficiently collect positive sample pages, we used only typical Japanese family names, hypothesizing that the presence of a researcher's name is independent of other statistical characteristics of a researcher's homepage.

It should also be noted that since the sample data were assessed manually, some entry pages of logical page groups that should be judged as positive might have been overlooked because they included very few or no clue words.

## 4.5 Property-based Keyword Lists

### 4.5.1 Problem Statement

The essential idea of the rough filtering is to utilize the information from the surrounding pages to gather as many researchers' homepages as possible. The search engine techniques used in the "Homepage Finding Task" [82] of TREC Web Track or in the NTCIR-4,5 WEB Navigational Retrieval Subtasks [81] [83] are not applicable to gathering researchers' homepages because we cannot use specific search keywords, such as person names and organization names, to collect all probable researchers' homepages and specific search targets are not provided.

Keywords, which can be either single words or compound words, are linguistic descriptors of documents [84] and often sufficiently informative to help human readers get a feel of the essential topics and main contents of the source documents [85]. They have also been used as features in many text-related applications such as text clustering [86], document similarity analysis [87] [88], and document summarization [89]. We therefore decide to use keywords in the rough filtering for gathering the candidate pages.

Content-based keywords are usually extracted from sample data by means of statistical techniques [63] [65], but for the following reasons this would be problematic in the case of property-based keywords:

1. Collecting a complete set of appropriate keywords is difficult because preparing a sample data set of a sufficient size is very expensive, especially for positive data.

2. The logical relations between the keywords and the definition of target data are unclear.

3. Tuning for improving the potential recall of unseen types of target data is impossible.

Fortunately, even though the styles and structures of homepages tend to differ greatly and the presentations of homepages are very divers, several basic information elements are common to the researcher's homepages defined in subsection 3.4.1. We therefore introduced **property-based keyword lists** representing the properties common to a majority of the target homepages, expecting that at least several of them are included in each target homepage and then the page can thus be identified as a target homepage. Note that the methods available for automatically extracting content-based keywords are not applicable for extracting property-based keyword lists where each of them contains a list of keywords grouped to the same property. We thus used an ad hoc method to create keyword lists for the present work.

We mainly used property-name-related terms. Property-value-related terms were used only when there were few of them; otherwise their maintenance required too much effort.

### 4.5.2 Procedure for Creating Keyword Lists

The conceptual procedure for creating the **keyword lists** is shown below. All the steps are executed manually.

**A.** Create keyword lists corresponding to properties common to researchers' homepages by studying the contents of positive sample data and the database structure of ReaD, etc.

**B.** For each keyword list select a small number of keywords from the contents of the positive sample data.

**C.** Add synonyms associated to the newly selected keywords by using a general Japanese dictionary.

**D.** Select new keywords from the contents of the positive sample pages that contain no keyword in any of the keyword lists.

**E.** Add those new keywords to the appropriate keyword list if there is one; otherwise use them to make a new keyword list. Split a keyword list in two if it starts to include keywords that are too diverse.

**F.** Repeat steps C through E until no more keywords remain to be selected.

Using this procedure in the current work, we created nine keyword lists in step A and by the time we had completed step F we had 12 keyword lists containing 86 keywords.

Each of the keyword lists was then assigned a type: either **organization-related** or **non-organization-related**. Keyword lists corresponding to the properties common to the members in the same organization were designated organization-related, while keyword lists corresponding to individual researcher's properties were designated non-organization-related. The types and meanings of these 12 keyword lists are listed in Table 4.1 along with some keyword examples[1]. Note that the actual keywords are in Japanese.

### 4.5.3 Evaluation

We confirmed the quality of the defined property-based keywords by using a T-test to evaluate the effectiveness of each selected keyword. The result indicated that 76 out of the 86 keywords (83.7%) were statistically significant at 95% confidence level ($p<0.05$).

We tried to extract other useful keywords by applying a T-test to each word whose document frequency was more than 5% in either positive or negative samples. Many of the extracted words, however, were simply university names, era names,

[1]The full list in the original language is given in Appendix A.

Table 4.1: Property-based keyword lists

| Type | Keyword list | Sample keywords* |
|---|---|---|
| non-organization-related | general word | research |
| | research topic | research topic, theme, etc. |
| | title | doctor, professor, etc. |
| | position | present position, duty, etc. |
| | history | biography, personal history, etc. |
| | achievement | paper, bibliography, etc. |
| | lecture | course, seminar, etc. |
| | academic society | academic society, regular member, etc. |
| organization-related | major | major, specialty, research field, etc. |
| | member | staff, member, etc. |
| | organization | university, institute, school, etc. |
| | section | section, department, etc. |

*Actual keywords are in Japanese.

place names, department names, and the like; only eight of the extracted words could have been considered as useful keywords. We do not use university names, for instance, as keywords because a complete list of the university names is hard to maintain over a long time, since they change rather frequently.

One of the reasons we use manually created keyword lists instead of independent keywords extracted automatically in the rough filtering is that doing so reduce the amount of memory needed to keep the document vectors. If we used the keywords extracted from the corpus there would be several thousand keywords and consequently several hundred bytes would be used for keeping each of the document vectors. In this case, the efficiency of the system will be a problem when the keyword matching is incorporated in the web crawler.

Please note that the essential role of the rough filtering is not making keywords and keyword lists but is utilizing the information from the surrounding pages by using the PGMs. The procedure for defining keywords and making keyword lists is at present a manual one since it is only a temporary method for confirming that

the proposed PGMs work well. Even though the experiment result on the rough filtering shows the effectiveness of the defined keyword lists and keywords, we are aware that we should make the defining procedure applicable to the homepages in other categories too. A survey of the results of the work on keyword extraction reveals that although some investigators have also tried to extract keyword lists automatically, the keyword lists are always given rather than extracted and any single set of keyword lists is not applicable to various categories. The keywords for each keyword list, however, can be obtained automatically by using the keyword extraction methods already available.

## 4.6 Page Group Models

To achieve high recall in gathering researchers' homepages, we need to take into consideration the structure in an individual logical page group.

Studying the relations between the entry page and the component pages of the homepages in logical page groups, we made the following observations: (1) the entry page and other component pages of a logical page group are always in the same site; (2) component pages of a logical page groups are always linked from the entry page of the logical page group; (3) component pages always link back to the entry page; (4) component pages are always in the same or lower directories in the directory subtree of the URL; (5) in/out-links from the upper directory in the directory path, the site top page and directory entry pages may contain some common information, such as organization-related information, that may not be contained in a logical page group.

We therefore consider only the local pages in same site and propose four simple page group models (**PGMs**) using (1) out-links to the same and the lower directories, (2) out-links to the upper directories, (3) in-links from the same and the upper directories, and (4) the site top and the directory entry pages in the same and the upper directories in the URL directory path.

In this section we present the concepts and definitions of the PGMs. Related notations are listed in Table 4.2, and the definitions of PGMs are listed in Table 4.3.

It should be noted that in all PGMs only the pages in the same site are considered.

Table 4.2: Notations

| Notation | Definition |
|---|---|
| $r$ | focused page |
| $P_{\mathrm{out-link}}(r)$ | set of pages linked from $r$ in the same site ($r$'s out-linked pages) |
| $P_{\mathrm{in-link}}(r)$ | set of pages linking to $r$ in the same site ($r$'s in-linked pages) |
| $P_{\mathrm{down}}(r, s, l)$ | set of pages in directories $s$ to $l$ levels lower in the directory subtree of $r$ ($r$'s lower pages) |
| $P_{\mathrm{up}}(r, s, l)$ | set of pages in directories $s$ to $l$ levels upper in the directory path of $r$ ($r$'s upper pages) |
| $P_{\mathrm{dir-ent}}(r, s, l)$ | set of entry pages of directories $s$ to $l$ levels upper in the directory path of $r$ |
| $P_{\mathrm{site-top}}(r)$ | set of $r$'s site top page(s) |
| $N_{\mathrm{Lod}}(r)$ | number of out-links from $r$ to the pages in the same directory and in the directory subtree of $r$ |

Note: The level of the same directory is defined as 0. $s$ and $l$ specify the range of directory levels.

### 4.6.1 SPM and SSM

Single page model (**SPM**) is a baseline model that uses keywords only in individual pages. Single site model (**SSM**) is another baseline model of the simplest PGM that uses keywords in all out-linked pages in the same site. We compared them with the proposed PGMs in order to evaluate the effectiveness of the proposed PGMs.

### 4.6.2 Simple PGMs

**PGM-Od** exploits all kinds of keywords in out-linked component pages in the lower levels of the directory subtree. **PGM-Ou**, **PGM-I**, and **PGM-U** exploit all kinds of keywords in component pages in the upper levels of the directory path; PGM-Ou is for out-linked pages, PGM-I is for in-linked pages, and PGM-U is for directory

Table 4.3: Definitions of PGMs and parameters

| **Model** | | **Description** | **Propagated pages** | **Tested Parameters** |
|---|---|---|---|---|
| SPM(baseline) | | A single page model: no keyword propagation is used. | – | |
| SSM(baseline) | | A reference page group model: all out-linked pages in the same site are used. | $P_{\text{out-link}}(r)$ | |
| Simple PGM | Od$(s,l)$ | A PGM based on out-links downward: out-linked pages in the URL directory subtree are used. | $P_{\text{out-link}}(r) \cap$ $P_{\text{down}}(r,s,l)$ | $s = 0,1$ $l = s..2$ |
| | Ou$(s,l)$ | A PGM based on out-links upward: out-linked pages in the directories included in the URL directory path are used. | $P_{\text{out-link}}(r) \cap$ $P_{\text{up}}(r,s,l)$ | $s = 0,1$ $l = s..4$ |
| | I$(s,l)$ | A PGM based on in-links upward: in-linked pages in the directories included in the URL directory path are used. | $P_{\text{in-link}}(r) \cap$ $P_{\text{up}}(r,s,l)$ | $s = 0,1$ $l = s..3$ |
| | U$(s,l)$ | A PGM based on directory entry pages: site top and directory entry pages of the directories in the URL directory path are used. | $P_{\text{site-top}}(r) \cup$ $P_{\text{dir-ent}}(r,s,l)$ | $s = 0,1$ $l = s..8$ |
| Modified PGM | Od@$\theta$ | Od with an additional condition on the number of out-links downward: Od is not used if there are too many out-links. | If $N_{\text{Lod}}(r) \leq \theta$, same as Od; otherwise, same as SPM. | $\theta = 5, 10, 20$; |
| | Ou#, I#, U# | Ou, I, and U: propagating only organization-related keywords. | Same as Ou, I, and U for organization-related keywords; for others, same as SPM. | |

Note: $r$ is a possible entry page; $s$ and $l$ specify the range of directory levels.

entry pages and site top pages.

Parameters used in each PGM are listed in Table 4.3. The parameters $s$ and $l$ specify the range of the directory levels to propagate the keywords from. The upper bounds of $l$ for PGM-Od, PGM-Ou, PGM-I, and PGM-U are respectively 2, 4, 3, and 8 because in the corpus all the pages in the lower and upper directories that are out-linked from any of the positive samples are within 2 and 4, all the in-linked pages of any of the positive samples are within 3, and the uppermost level of directory entry pages of all the positive samples is 8.

### 4.6.3 Modified PGMs

PGMs usually propagate many keywords irrelevant to the researcher to the virtual entry page and consequently include many noise pages. We therefore introduce modified PGMs to reduce such noises while ensuring that useful keywords are propagated.

**PGM-Od@$\theta$** is a modified PGM derived from PGM-Od with the intention of excluding irrelevant pages introduced by PGM-Od. The modification was based on the observations that one of the noise sources is groups of many pages mutually linked within a directory and that an entry page having many out-links within the directory subtree always contains sufficient keywords.

**PGM-Ou#**, **PGM-I#**, and **PGM-U#** are respectively modified PGMs derived from PGM-Ou, PGM-I, and PGM-U. Because we observed that non-organization-related keywords are not included in the upper directory hierarchies, only organization-related keywords are propagated in these modified PGMs.

Note that because the component pages in the same and lower directories are the most informative, all kinds of keyword lists will be propagated through PGM-Od@$\theta$.

### 4.6.4 PGM Combinations

Since a target page has a wide range of variation but each simple or modified PGM can utilize only a part of the available component pages, no single PGM can collect sufficient information. We therefore use combinations of modified PGMs with the

intention of utilizing as much useful information as possible from the component pages but introducing as little noise pages as possible.

## 4.7 Experiment Results

This section presents experiment results showing that more difficult pages are found when using PGMs than that are found when using the SPM.

Using the 100GB data set described in Section 4.4, we experimented on PGMs with various parameters. The results show that the amount of pages extracted can be reduced to an allowable level by the rough filtering using PGMs even though using PGMs always introduce a lot of noise. The effectiveness of using PGMs rather than the SPM is shown by additionally assessing the candidate pages that were output by the proposed method but were judged as negative in the first manual assessment.

To find out whether the rough filtering using PGMs is applicable to a larger data set, we also apply it to the 1.36TB data set (NW1000G-04).

The results from various experiments are shown below in several graphs. All the graphs include SPM and SSM results for comparison. The $x$-axes are the page amount $n_c(i)$, specifically, the number of pages in the corpus that scored at least $i$ ($1 \leq i \leq 12$). The $y$-axes are recall defined by $n_p(i)/N_p$, where $N_p$ is the total number of positive sample data and $n_p(i)$ is the number of positive sample data that scored at least $i$.

The upper-right-most data in each plotted curve corresponds to a threshold score of 1, and every next one corresponds to a threshold score incremented by 1. In general, a higher recall and a smaller amount of pages indicate better performance, but we put priority on the recall.

Note that our analysis is accurate on condition that the samples are randomly selected from the population, and note that it does not take into account manual assessment errors (false-positives and false-negatives).

Since the amount of positive sample data is not large, the confidence intervals of the recall values observed in the following experiments should be considered.

Since confidence intervals are considered at a very high recall range, a binomial distribution rather than a normal distribution should be used. Figure 4.2 shows the 80%, 90%, and 95% confidence intervals calculated with a binomial distribution for the observed recall value $x$ with the sample size fixed to 426. Some values of interest are also listed in Table 4.4. To assure at least 98% recall (lower confidence limit) at the 90% confidence level, the observed recall must be a little higher than 99%.

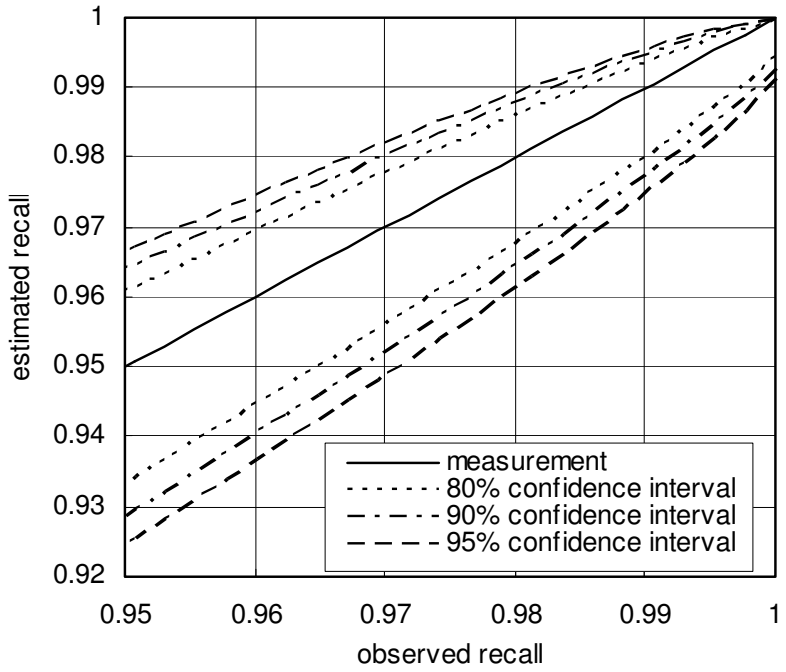


Figure 4.2: Confidence intervals of recall for 426 samples.

Table 4.4: Confidence intervals of observed recalls

| **Confidence level** | **95%** | | | **90%** | | |
|---|---|---|---|---|---|---|
| Observed recall | 97.0 | 98.0 | 99.0 | 97.0 | 98.0 | 99.0 |
| Upper confidence limit | 98.2 | 98.9 | 99.6 | 98.0 | 98.7 | 99.5 |
| Lower confidence limit | 94.8 | 96.0 | 97.6 | 95.2 | 96.3 | 97.9 |

Note: Recalls are in percentages.

### 4.7.1 Comparison of Simple PGMs

We first experimented on individual simple PGMs with typical parameters in order to understand their basic performances. The results for PGM-Od(0,2), PGM-Ou(0,3), PGM-I(0,3), and PGM-U(0,3) are shown in Figure 4.3. It shows that all the simple PGMs are inferior to SPM (in terms of page amount) because a lot of noise is introduced by the keyword propagation but is superior to SSM.

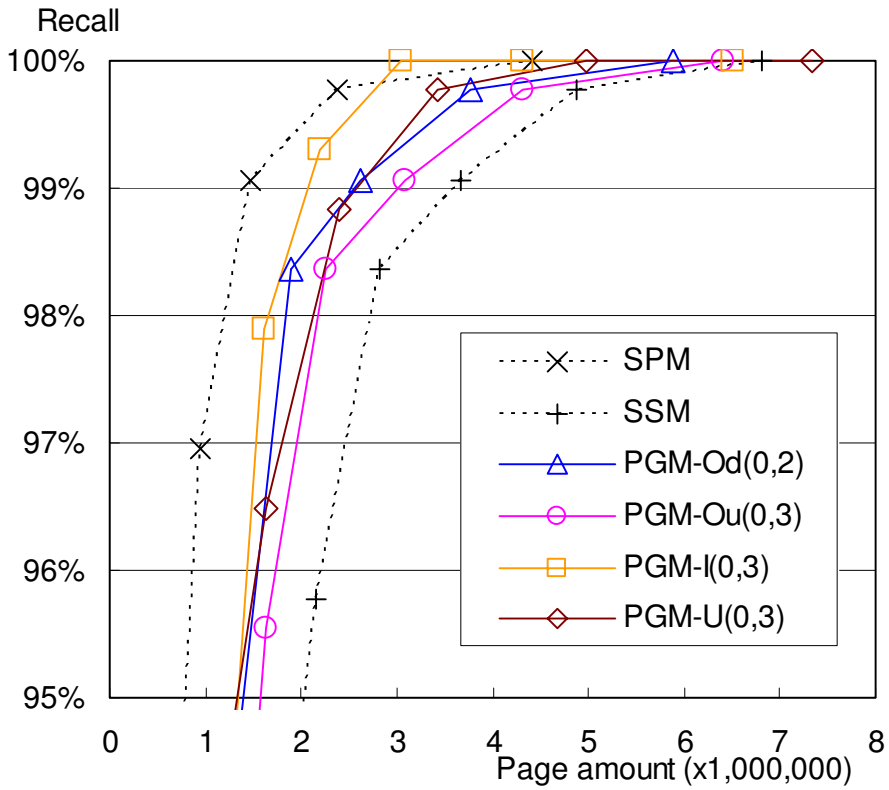


Figure 4.3: Performance of simple PGMs.

### 4.7.2 Effects of Modified PGM-Od

We next experimented on simple and modified PGM-Od's with several $s$, $l$, and $\theta$ values. Some typical results are shown in Figure 4.4. Results are shown only for cases where $l = 2$ because the change of $l$ made almost no difference for each combination of $s$ and $\theta$. For $s = 1$, only results for PGM-Od(1,2) are shown because modified PGM-Od results for every $\theta$ were almost the same as for simple PGM-Od.

The figure indicates that if we select $s = 1$, the page amount is nearly the same

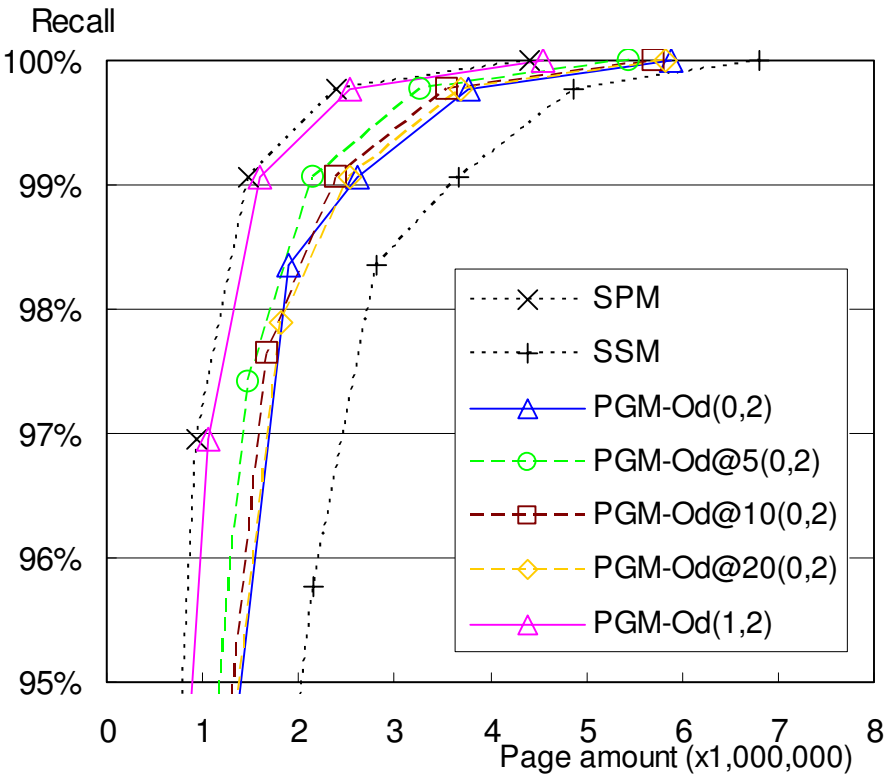


Figure 4.4: Performance of simple and modified PGM-Od's.

as that for SPM, whereas at each threshold score the recall value is the same for SPM. This implies that almost all non-organization-related keywords are collected from within the same directory. Thus, $s = 1$ is useless for PGM-Od. On the other hand, focusing on the recall area around 99%, we can see that if we select $s = 0$ the page amount for simple PGM-Od increases by 80% over that for SPM whereas the page amount for modified PGM-Od is only 50% over that for the SPM.

Although smaller $\theta$ tends to result in a smaller amount of pages, since PGM-Od is the only PGM that propagates non-organization-related keywords, the parameters should be carefully selected and will be further investigated in subsection 4.7.4.

### 4.7.3 Effects of Modified PGM-Ou, PGM-I, and PGM-U

We then experimented on modified PGM-Ou, PGM-I, and PGM-U, and compared them with the corresponding simple PGMs with typical parameters.

The results for PGM-Ou and PGM-I are shown in Figures 4.5 and 4.6. Results

are shown only for cases where $l = 3$ because the changes of $l$ made almost no difference for each $s$. The results for PGM-U are shown in Figure 4.7, where results only for cases where $s = 0$ and $l = 3$ are shown because the changes of $l$ made almost no difference for each $s$ and the changes of $s$ caused only small shifts of the corresponding plots.

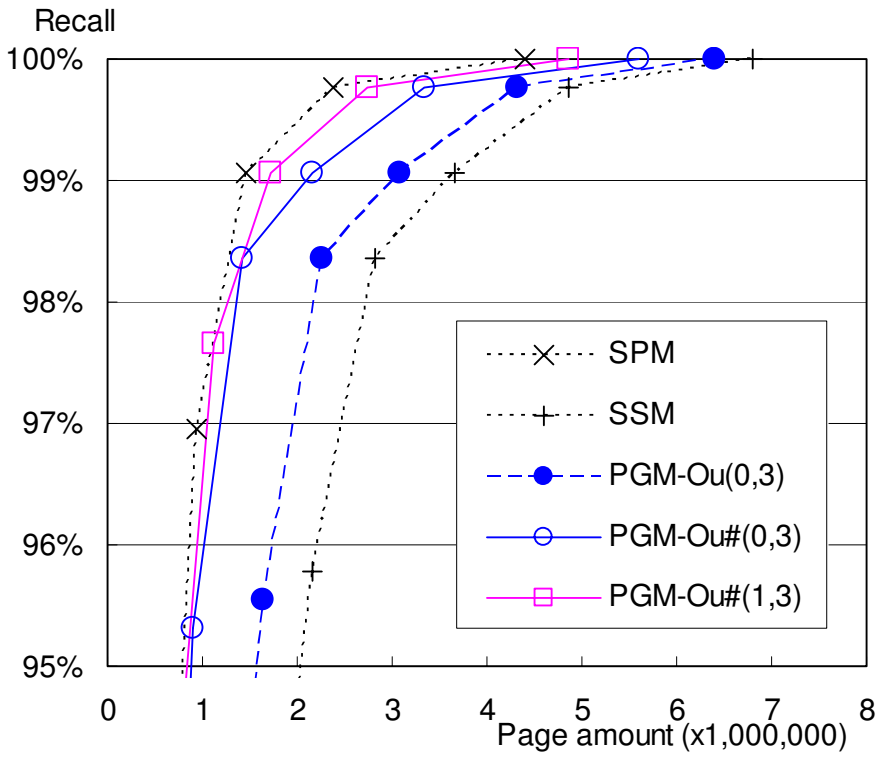


Figure 4.5: Performance of simple and modified PGM-Ou's.

Focusing on the recall area around 99%, we can see that although with each simple PGM the page amount increases by 40% to 120% over that for SPM, modified PGMs can reduce the page amount to nearly the same level as that for SPM.

### 4.7.4 Effects of PGM Combinations

We also experimented on combinations of PGMs with several promising parameter sets.

Our parameter selection for each PGM was based on the following policy: if the page amount difference between two parameter sets is small in the recall area around 99%, then the one that collects keywords from more pages should be selected.

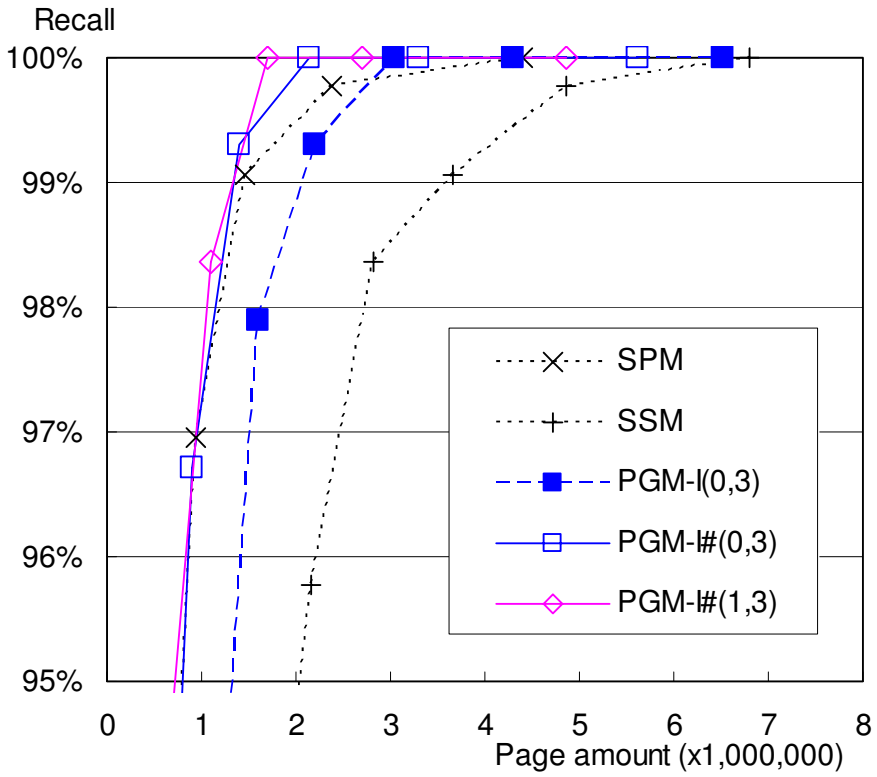


Figure 4.6: Performance of simple and modified PGM-I's.

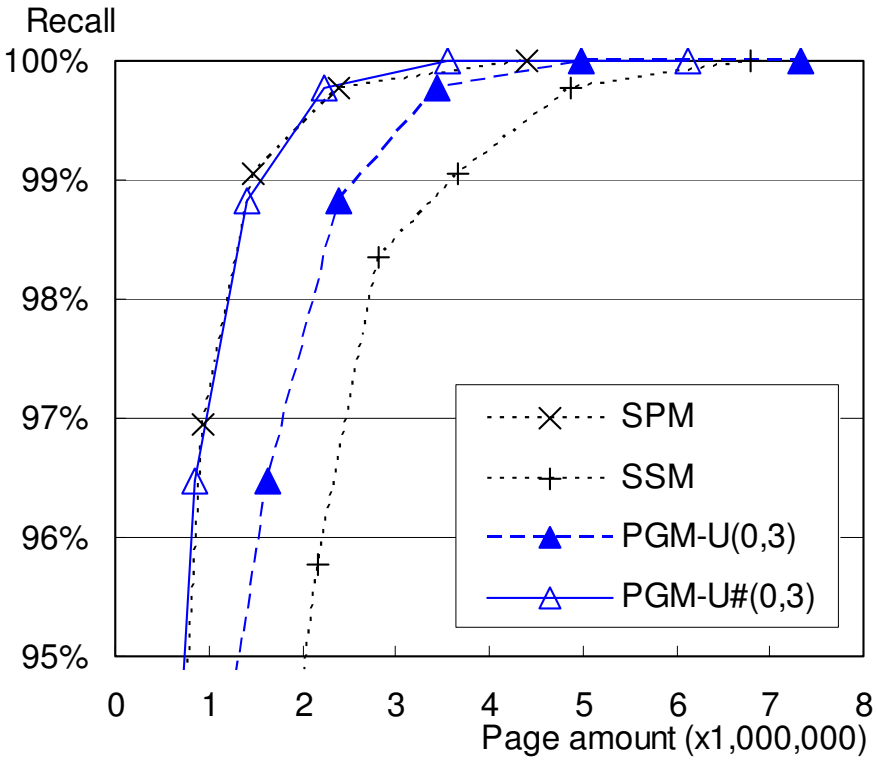


Figure 4.7: Performance of simple and modified PGM-U's.

Furthermore, for PGM-Od, we selected $s = 0$ for the reason mentioned in subsection 4.7.2, although modified PGM-Od still collected a rather large amount of noise pages.

Since pages in the same directory have many chances to be used as keyword sources, combinations of $s = 1$ as well as $s = 0$ were tested for the other PGMs.

Figure 4.8 shows the results for three well-performing combinations of PGMs. We refer to them hereafter as follows:

PGM-C1: PGM-Od@5(0,2),Ou#(1,3),I#(0,3),U#(0,3)

PGM-C2: PGM-Od@10(0,2),Ou#(1,3),I#(0,3),U#(0,3)

PGM-C3: PGM-Od@20(0,2),Ou#(1,3),I#(0,3),U#(0,3)

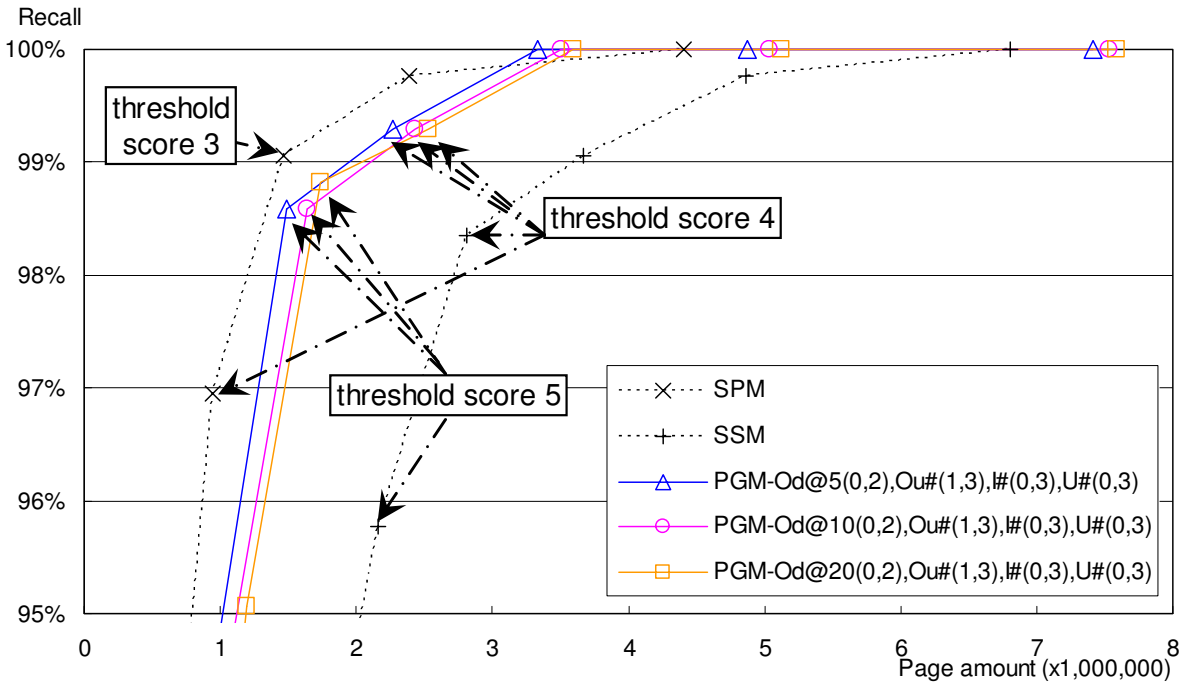


Figure 4.8: Performance of selected PGM combinations.

Each of them uses all four modified PGMs with the same parameters except for the $\theta$ of PGM-Od. As $s = 0$ is used for PGM-Od, $s = 1$ is selected for PGM-Ou. All the other parameters were eventually the same for all combinations. Note the scale of the $x$-axis is 1.5 times larger than that in the other figures.

The results show that even the best performance obtained with PGM-C1 is inferior to that obtained with the SPM in all the recall ranges except for 100%. On the other hand, it is also shown that the proposed method reduced the amount of pages to a reasonable level despite its use of PGMs.

Since, as mentioned in Section 4.4, some true positive data is overlooked in the manual assessment, the effectiveness of the proposed method is not clearly shown by the results plotted in Figure 4.8. This issue is further investigated in the following subsection.

### 4.7.5 Ability to Find Difficult Pages

We evaluated the ability of the proposed method to find positive pages that were overlooked in the manual assessment by using additional assessments to confirm the effectiveness of the proposed method.

When the Jname data that scored less than 3 with SPM but scored at least 4 with PGM-C3 were assessed again, 13 new positive pages were found. For each of these pages, we checked the scores obtained with SPM and PGM-C1 through PGM-C3. The number of pages overlooked at each score level is listed in Table 4.5.

Table 4.5: Overlooked positive pages at each score

| | | **Score for SPM** | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | Total |
| **Score for PGM-C1 to PGM-C3** | 4 | 1 | 2 | 0 | 3 |
| | 5 | 2(1) | 5 | 0 | 7(6) |
| | 6-12 | 1 | 1 | 1 | 3 |
| | Total | 4(3) | 8 | 1 | 13(12) |

**Note**: Each listed value is the number of positive pages. All the numbers are same for PGM-C2 and PGM-C3. The numbers in parentheses are those for PGM-C1.

The values listed in Table 4.5 show the ability of the proposed method to find positive pages that cannot be gathered by SPM even if we select the threshold score

3 so that the recall is more than 99% for the manually assessed positive samples. An analysis of the results is given in the following section.

### 4.7.6 Applicability to a Larger Data Set

We applied the rough filtering to the larger (1.36TB) data set by using the following procedure, which is similar to the procedure we used with the 100GB data set. Approximate computational complexities of the processing steps are shown at the end of each line. $N$ is the number of the web pages in the corpus.

1. Search keywords in the full text index: O(log $N$)
2. Using the link list files, make a page list for each keyword list for each PGM: O($N$ log $N$)
3. Merge the page lists of all PGMs for each keyword list and calculate the scores: O($N$ log $N$)
4. Select pages that scored at least the threshold value: O($N$).

When we incorporate the rough filtering in the crawling process, step 1 will require computation of O($N$). Anyway, the overall processing cost for the rough filtering is of O($N$ log $N$).

The same parameters of PGMs used within the 100GB data set were also applied to the larger data set. The threshold number of out-links for PGM-Od was set at 20, and the threshold score for gathering candidate pages was set at 4. The results obtained with each of the data sets are listed in Table 4.6.

Table 4.6: Comparison of pages output from the rough filtering for two data sets

| **Data set** | **Total amount** | **Output amount** | **Output proportion** |
|---|---|---|---|
| 100GB data set | 11,038,720 | 2,530,850 | 22.9% |
| 1.36TB data set | 95,870,352 | 14,128,826 | 14.7% |

Note: threshold number of out-links for PGM-Od is 20 and the threshold score is 4.

The experiment results show that the rough filtering reduced the output pages even more for the larger data set. As we have not assessed the correctness of the output, however, we cannot conclude that the accuracy of the output is stable.

## 4.8 Considerations

We obtained 480 positive pages when we randomly sampled 1% of the pages output from the rough filtering and manually assessed them. Thus, taking the sampling error into consideration, we estimated the total number of positive pages in the corpus to be $480/1\% = 48000(\pm 4380$ pages at 90% confidence level).

Table 4.7 summarizes the results for PGM-C1, PGM-C2, and PGM-C3. The precision was calculated by dividing the estimated total number of positive pages by the amount of pages output by the rough filtering. The precision values are rather low comparing to the precisions obtained using state-of-the-art web page classification methods. Considering the very high recall, however, we think the performance of the rough filtering is fairly good. The "Reduction ratio" is the rough filtering output as a percentage of the corpus size. The values are not satisfactory but show that the processing cost of the following processes can be greatly reduced (less than 23% for the 100GB data set and 15% for the 1.36TB data set).

Table 4.7: Summary of experiment results

| PGM | Threshold score | Recall | Precision* | Page amount | Reduction ratio |
|---|---|---|---|---|---|
| PGM-C1 | 4 | 99.3% | 2.12% | 2,265,478 | 20.5% |
| | 5 | 98.6% | 3.24% | 1,482,980 | 13.4% |
| PGM-C2 | 4 | 99.3% | 1.98% | 2,429,250 | 22.0% |
| | 5 | 98.6% | 2.93% | 1,635,703 | 14.8% |
| PGM-C3 | 4 | 99.3% | 1.90% | 2,530,850 | 22.9% |
| | 5 | 98.8% | 2.76% | 1,738,404 | 15.7% |

*Precisions are estimates.

When the 13 newly found positive pages are taken into account, SPM's recalls at threshold scores 2 and 3 should be corrected from 99.8% (425/426) to 97.0%

(426/439) and from 99.1% (422/426) to 96.1% (422/439). Comparing these values with the recalls of the proposed methods at threshold scores 4 and 5 reveals that the proposed methods obviously (at 95% confidence level) outperform SPM. Furthermore, four positive pages cannot be gathered with SPM even if the threshold score is set to 1. This implies SPM can hardly achieve the target recall with a feasible amount of pages.

A failure analysis on all three pages that scored only 3 with PGM-C1 through PGM-C3 revealed the following. All of them were researchers' introduction pages from the same site officially provided by a university department. Their page styles were similar and they contained little information, scoring only 2 with SPM. Although they had hyperlinks to the researchers' personal homepages, our method could not find them because they were in separate sites.

Finally, as there are trade-offs between the recall and the page amount, it is difficult to say in general which of PGM-C1, PGM-C2, and PGM-C3 is the best. To guarantee that the overall recall will be more than 98%, we should set the threshold score to 4 considering the number of sample pages. We selected PGM-C2 as the most appropriate for the current goal because the recall at threshold score 4 is the same for PGM-C2 and PGM-C3. The experiment results show that even PGM-C3 was used, the number of pages output from the rough filtering was not more than 23% of the pages in the 100GB data set.

The experiment results obtained when applying the rough filtering to the larger, 1.36TB, data set show that the number of pages output from the rough filtering is only 15% of the pages in the data set. Since it is much less than the percentage of pages output from the rough filtering of the 100GB data set, we can conclude that the rough filtering is especially effective for larger web data sets.

We did not assign relative weights to each of the keyword lists used in the rough filtering, but we did notice that the importance of each keyword list was not the same. The rough filtering might work better if the keyword lists were weighted according to their importance, but this remains to be investigated.

## 4.9 Conclusion

We described the rough filtering method for gathering candidate researchers' homepages from the web comprehensively while gathering as few pages as possible by using property-based keyword lists combined with four page group models. Two original key techniques were introduced to reduce irrelevant keywords to be propagated by exploiting the mutual relations between the contents and structures of pages in a logical page group. One is to introduce a threshold on the number of out-linked pages in the same and lower directories, and the other is to consider two types of keyword lists and to propagate only one type (organization-related keyword lists) from the upper directories.

We evaluated the method by comparing its performance with that of a single-page-based method in experiments using a manually created sample data set with various parameters. It successfully reduced the increase in the amount of gathered pages to an allowable level despite its use of the page-group-based method, which generally causes much more noises. It was also able to gather a significant number of the target pages that a single-page-based method could not gather.

The method is effective for narrowing down the candidate pages for the web data sets, especially larger ones. It is considered to have fulfilled the goal set for the rough filtering.

# Chapter 5

# Accurate Classification

## 5.1 Introduction

The accurate classification classifies the input data into three categories—assured positive, assured negative, and uncertain—with the required recall and precision and does so while keeping the amount of data classified uncertain as small as possible.

A preliminary experiment using SVM$^{light}$ to classify the researchers' homepages in a sample data set by using only features based on content words of the individual pages resulted in 59.6% recall and 90.15% precision. One reason for such a low recall is probably that some of the target homepages contain links to component pages themselves contain little textual content. It is therefore essential to exploit information contained in the component pages of the logical page group. Although existing web page classification techniques exploit a variety of information related to a target page (e.g., textual content, html tag, page structure, page layout, URL, directory structure, anchor text, and hyperlink structure), few of them exploit information contained in the component pages and even fewer exploit that information in combination with other types of information.

In this chapter, we discuss a web page classification method that uses Support Vector Machine (SVM) and not only uses features that are obtained from the textual contents in the target page and its surrounding pages but also uses them in combination with the relative locations of the pages. Here the relative location is given by a combination of page relation and directory level, where the page relation

is either (1) in-linked pages, (2) out-linked pages, or (3) the directory entry pages (including site top pages) in the directory path to the site root, and the directory level is either (a) in the same directory, (b) in the upper directories in the URL path, or (c) in the lower directories of the directory subtree. We use two types of textual features: plain-text-based and tagged-text-based.

Although the classification performance can be notably improved with the proposed method, it is still impossible to make such a high-quality homepage collection as required for our application solely by computer processing. Human involvement is therefore indispensable in overcoming the gap between the requirement and the achievable performance, and the problem arising here is how to reduce the number of web pages requiring human assessment for quality assurance. We approached this problem by devising a recall-assured classifier and a precision-assured classifier and using them in combination.

Using the sample data collected from NW100G-01, a 100GB web data set, we discuss the effectiveness of the classifiers by comparing it with the baseline which using only the features obtained from the plain-text-based content in the target page.

In addition, to evaluate the applicability of the proposed classification method, we apply the feature sets of the well-performing classifiers to the Web->KB data set and compare the performance obtained in related research. We also analyzed examples of the classification results in order to better understand the effectiveness of the classifiers.

The rest of the chapter is organized as follows. Related work is introduced in Section 5.2, the composition of the accurate classification which consists of two kinds of classifiers is explained in Section 5.3, and section 5.4 presents the main characteristics of the classifiers, including plain-text-based and tagged-text-based features, feature subsets, and feature sets. Section 5.5 shows the experiment results and Section 5.6 discusses them. Section 5.7 concludes the chapter by summarizing it briefly.

## 5.2 Related Work

The method we will discuss in this chapter could be considered as a web page classification method by exploiting the content information in the surrounding pages and taking into account their locations relative to the target page.

Classification has numerous applications in the hypertext and semi-structured data domains. The methods for web page classification generally try to exploit various types of web-related information sources for extracting features such as textual content, html tag, URL, and the directory/hyperlink structure. The method using only the features based on the textual content of the target page is the simplest one and can be used as a baseline for performance comparisons.

There has been some research on classification methods using hypertext and its neighboring text in order to utilize some of the contents of the globally related pages in addition to textual-content-based features. For example, Chakrabarti et al. proposed a method for enhanced hypertext categorization using hyperlinks [75]; Sun et al. conducted a research on a SVM-based classifier using plain-text-based features as well as html tags (title contents) and hyperlinks (anchor texts) as web page features [72]; Glover et al. devised a method for classifying and describing web pages by using web structure [26]; and Chau proposed a machine-learning-based approach that combines features related to web content and web structure for the classification of a large collection with only a small number of training examples [76].

Some researchers have used features utilizing the URLs as well as the contents of target pages. Shih and Karger, for example, proposed some new features using URLs and table layouts for web classification tasks, including content recommendation and ad blocking [32], and Kan and Thi proposed a fast web page classification method using only URL features [74].

Other researches have used features utilizing local surrounding pages as well as the contents of target pages. Sun and Lim proposed an iterative web unit mining method for finding and classifying web units of web pages [48], and Masada et al. proposed a method for improving the web search performance obtained with a Vector Space Model exploiting local link information by means of a new web page clustering algorithm [69].

The abovementioned research works used various kinds of web-related features and improved the performances of the classifications to various degrees. All of these information sources except the last two are used to identify characteristic features of the target pages and are effective in reducing noises. The last two ones, in contrast, are used to collect information dispersed in the surrounding pages by exploiting the local link structure and the context about where the page is placed. They are effective with regard to comprehensiveness but tend to increase noises. In addition, information about the global structure, such as anchor text or global link structure, is available only for popular web pages and is not available for very new pages or dynamically generated pages.

Since the comprehensiveness is a key factor for assuring the quality of a web page collection, in the accurate classification we exploit the features on the contents in surrounding pages by considering local link structures. This, however, generally introduces a lot of noises and we try to reduce it by not merging the features of surrounding pages but concatenating them independently and using them all together in the classification so that the contexts corresponding to the relative locations are represented. We introduce several feature subsets on the surrounding pages, in combination with the connection types (in-linked, out-linked, and directory entry pages) and relative URL levels (in the same, upper, or lower directories). Through this way, various information sources can be exploited independently.

Besides exploiting features of plain-text-based content, we propose a very simple method for extracting the minimum amount of tagged-text-based features by exploiting the characteristic content words in the pages. We do this expecting to extract property names and/or values that are useful to our task. Features from other information sources can be easily combined with our method later.

None of the abovementioned research has assured the high recall and/or high precision required by practical applications, we therefore decided to use a recall-assured classifier and a precision-assured classifier in combination in order to satisfying requirements of practical applications. We also expect that the amount of human assessment needed to satisfy the performance requirements can be reduced by using the two kinds of classifiers in combination.

## 5.3 Classification Scheme

### 5.3.1 Composition of the Three-way Classifier

Two compositions of a three-way classifier using two kinds of binary classifiers, a **recall-assured classifier** and a **precision-assured classifier**, are shown in figure 5.1 and 5.2. One uses them in parallel and the other uses them in series.

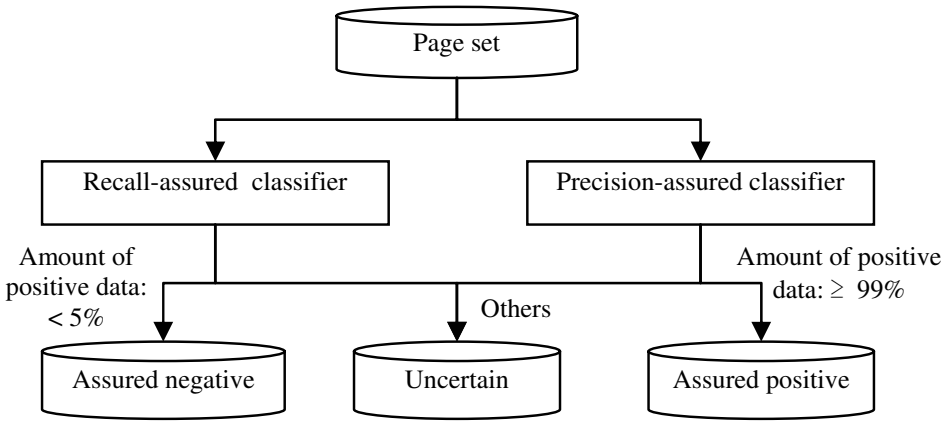


Figure 5.1: Composition of the three-way classifier with a recall-assured classifier and a precision-assured classifier in parallel.

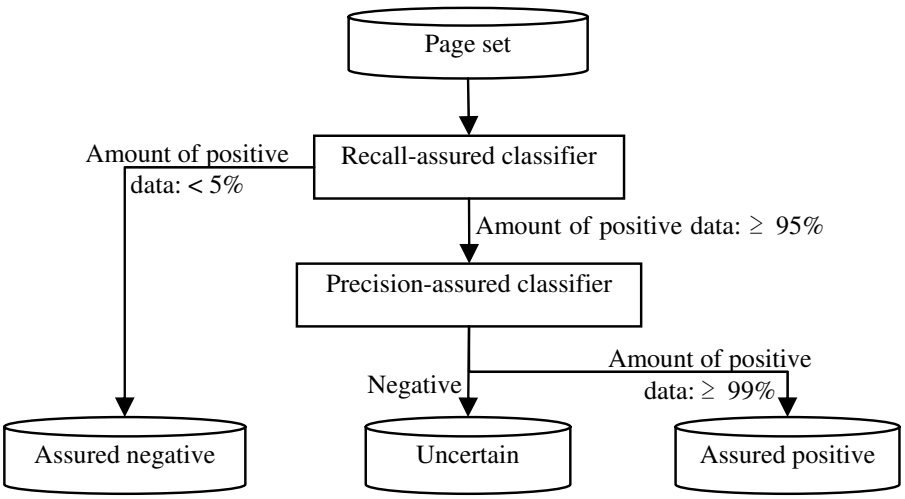


Figure 5.2: Composition of the three-way classifier with a recall-assured classifier and a precision-assured classifier in series.

The recall-assured classifier is used for filtering out as much negative data as possible from the input pages under the condition that the recall is at least 95%. The pages labeled negative are automatically classified as negative and are not processed any more.

The precision-assured classifier is used for collecting as much positive data as possible under the condition that the precision is at least 99%. The pages labeled positive are automatically classified as positive and are not processed any more.

The pages not labeled positive or negative are classified as uncertain and need to be assessed manually later in order to assure the required quality. It is expected that the number of uncertain pages can be reduced to a certain degree by improving the performances of the classifiers.

Note that the outputs of the two compositions are exactly the same by logics under the condition that the pages labeled positive by the precision-assured classifier are always labeled positive too by the recall-assured classifier, and the condition holds practically at all times. The discussion in this chapter thus applies to both of the compositions. Their processing efficiencies, however, are different and will be discussed in the following chapter.

### 5.3.2 Tool

Support Vector Machine (SVM) is used to learn classification functions from a set of labeled training data. SVM-based classifiers have shown promising results in text classification. They are both efficient and effective in many kinds of classification problems. Joachims [90], for example, have summarized the suitability of SVM for text classification, and Bekkerman et al. [71] reported that SVM used together with MI feature selection method outperforms other classification methods. Other researchers have also compared different classification methods and found that SVM always works well [20] [91].

One of the distinctive characteristics of our method is we use concatenated features rather than merged features and the feature space is very sparse. Another is that we concatenate up to nine feature subsets from individual surrounding pages to compose the feature set and the feature space tends to become high-dimensional.

Since SVM is especially good at the classification problems with high-dimensional feature space and sparse document vectors [90], we decided to use it as the method for the accurate classification. In the present work we used the SVM$^{light}$ package with the linear kernel.

As will be described in next section, we use two parameters $c$ and $j$ for tuning the classifiers: $c$ is the tradeoff between training error and margin, and $j$ is the cost-factor by which the training error on positive examples out-weights errors on negative examples[1].

### 5.3.3 Tuning Method

For ordinary binary classification, the F-measure is often used for evaluating the performance of a classifier. As it is a harmonic mean of recall and precision, however, it does not indicate the classifier's performance at a specific recall or precision. We therefore needed to use a new tuning method.

As two kinds of classifiers are used in the proposed scheme, it is expected that the performance obtained by selecting features and tuning parameters for each classifier independently will be better than that obtained by adjusting a single generally tuned classifier. We therefore tested and compared the two tuning methods as follows:

1. ***offset*-based tuning method**

   SVM outputs directed distances from a learned separating hyperplane to respective document vectors, and each document is labeled according to the sign of its "*distance-offset*". Although *zero* is usually selected for the *offset*, recall (or precision) can be adjusted by moving the *offset*. In the experiments using the *offset*-based tuning method, we first selected features and tuned $c$ and $j$ parameters to maximize the F-measure and then adjusted the *offset* to fit to the condition on recall or precision.

2. ***c-j-option*-based tuning method**

[1] SVM$^{light}$ Support Vector Machine. http://svmlight.joachims.org/.

Another way to adjust recall or precision is to tune the combination of $c$ and $j$ parameters to achieve, for the recall-assured classifier, the best precision at the given recall, and for the precision-assured classifier, the best recall at the given precision.

Tuning the $c$ and $j$ parameters requires many trials and errors because their behavior is nonlinear. Conversely, adjusting *offset* is straightforward and simple because of its linearity. Moreover, when a constraint on recall or on precision is given, tuning $c$ and $j$ becomes much more complicated because they interfere with each other.

## 5.4 Classifiers

### 5.4.1 Plain-text-based Feature Word Selection

Many previous papers have reported the necessity of using feature selection to improve generalization and avoid over-fitting even when using SVM. In addition, for the reasons of both efficiency and efficacy, feature selection is often used when applying machine learning methods to text categorization, as has been done by Joachims [90], Dumais [19], and Craven [46].

In our research, on only the plain-text-based features of the sample pages (which will be described in subsection 5.5.2), the number of feature words was more than 30,000. Therefore, if we used features on various surrounding pages concatenated without selection, both the over-fitting and the efficiency of the classification would have been a big problem.

We used the following procedure to select the **plain-text-based feature** words:

1. All the sample pages are processed by "Chasen"[2].

2. The words output from "Chasen" are selected under both the following two conditions:

[2]A morphological tool. http://chasen.naist.jp/hiki/ChaSen/.

- The POS (part of speech) tag of the word is "noun" but not "person names" or the POS tag is "unknown";
- The document frequency of the word is at least 5.

3. The mutual information (MI)[3] [92] measure $I(C, W_i)$ defined as follows is calculated for each word.

   Let $W_i$ be a random variable indicating whether word $w_i$ is present or absent in a document, and $v_i \in \{w_i, \bar{w_i}\}$ be the values it takes on. Let $C$ be a random variable taking values of two class values, $c \in \{Pos., Neg.\}$. Then, the mutual information [93] is

$$I(C, W_i) = \sum_{v_i \in \{w_i, \bar{w_i}\}} \sum_{c \in C} Pr(c, v_i) log(\frac{Pr(c, v_i)}{Pr(c)Pr(v_i)})$$

4. Words are selected from the largest MI down to a given number.

To determine the feature word number $n$, we did a preliminary experiment on $n$ values ranging from 500 to 3,200 (at intervals of 100) using SVM$^{light}$ with its default setting. The results showed that the performance of the classifier was almost the same at feature word numbers from 2,000 to 3,200 but deteriorated when the feature word number was less than 2,000. Consequently, we used the top 2,000 words for plain-text-based features.

### 5.4.2 Tagged-text-based Features

We proposed a very simple method to extract tagged-text-based features for exploiting the characteristic content words in the pages, with the expectation of extracting property names and/or values useful to our task.

**Tagged-text-based features** are extracted from short text segments marked up by HTML tags with the following procedure.

1. Every "*text*" satisfying both the following conditions is extracted from the positive and negative sample data described in Chapter 3:

[3]The mutual information of two random variables is a quantity that measures their mutual dependence. For details, refer to [92] and http://en.wikipedia.org/wiki/Mutual_information.

- Matches to the pattern ">text<" or "<img ... alt="*text*" ...>";
- The length of the "*text*" is no more than 16 bytes omitting spaces.

2. The extracted text segments are processed by "Chasen".
3. The extracted word is selected as a tagged-text-based feature word when the following conditions hold:
   - Its POS tag is "noun" but not "person name" or its POS is "unknown";
   - Its positive (or negative) document frequency is not less than 1% of the total amount of positive (or negative) sample data.

Note the first pattern we used for extracting tagged-text-based features treats all kinds of the tags, including start tags and end tags, equally because there are variety of tag uses for the layout of web pages (headers (<H1>, etc.), tables, definition lists, font colors, etc.).

The total number of tagged-text-based feature words extracted from positive and negative sample data was 1,026. Since this is not large and all the words were expected to be the properties of the topic of interest, we used all of them as tagged-text-based feature words but not apply feature word selection to them.

### 5.4.3 Feature Subsets and Feature Sets on Surrounding Pages

The surrounding pages are grouped according to connection types (in-link, out-link, and directory entry) and relative URL hierarchy (same, upper, or lower in the directory hierarchy). Their definitions and groups are listed in Tables 5.1 and 5.2.

Each group of surrounding pages has its own potential meaning in a logical page group. Pages in G6, for example, might represent component pages having back links to the entry page, and pages in G9 might be entry pages of the organization the researcher belongs to.

A **feature subset** F$x$ is generated from the pages in group G$x$. The feature subsets are further concatenated conceptually to compose the **feature set** of a classifier. Typical examples of feature sets are listed in Table 5.3. Figure 5.3 illustrates

Table 5.1: Definitions of surrounding pages

| Notation | definition |
|---|---|
| $r$ | target page |
| $P_{\mathrm{out-link}}(r)$ | set of pages linked from $r$ in the same site ($r$'s out-linked pages) |
| $P_{\mathrm{in-link}}(r)$ | set of pages linking to $r$ in the same site ($r$'s in-linked pages) |
| $P_{\mathrm{dir-ent}}(r)$ | set of directory entry pages in $r$'s directory path |
| $P_{\mathrm{site-top}}(r)$ | set of $r$'s site top page(s) |
| $P_{\mathrm{same}}(r)$ | set of pages in the same directory as $r$ ($r$'s same directory pages) |
| $P_{\mathrm{low}}(r)$ | set of pages in the lower directory subtree of $r$ ($r$'s lower pages) |
| $P_{\mathrm{up}}(r)$ | set of pages in the upper directory path of $r$ ($r$'s upper pages) |

Table 5.2: Groups of surrounding pages

| | $r$ | $P_{\mathrm{out-link}}(r)$ | $P_{\mathrm{in-link}}(r)$ | $P_{\mathrm{dir-ent}}(r) \cup P_{\mathrm{site-top}}(r)$ | Merged |
|---|---|---|---|---|---|
| $P_{\mathrm{same}}(r)$ | G1 | G2 | G5 | G8 | GS |
| $P_{\mathrm{low}}(r)$ | | G3 | G6 | | GL |
| $P_{\mathrm{up}}(r)$ | | G4 | G7 | G9 | GU |
| Merged | | GO | GI | GD | |

the way the feature sets are composed. For example, the feature set u-1 is made by concatenating the feature subsets on pages in G1, G4, G7, and G9. It should be noted that each feature set contains the feature subset on the target pages of G1.

Comparing Table 5.1 and Table 4.2, it can be seen that the rough filtering and the accurate classification use similar basic concepts of surrounding pages considering the similar page group structures. There are two minor differences: for the rough filtering, the directory levels of the pages to be propagated by PGMs are limited taking into account the efficiency, and the threshold number of out-links in the same and lower directories are introduced for reducing the noises caused by the use of PGMs; while for the accurate classification, we do not use such ad hoc techniques because we can obtain better performance by using more systematic techniques.

It should be noted that, in general, the number of feature words need not be

Table 5.3: Typical feature sets

| Name | Description | Surrounding page groups used |
|---|---|---|
| baseline | Target page only | G1 |
| o-1 | Out-linked pages | G1, G2, G3, G4 |
| i-1 | In-linked pages | G1, G5, G6, G7 |
| o-i-1 | In-/out-linked pages | G1, G2, G3, G4, G5, G6, G7 |
| o-i-2 | Merged in-/out-linked pages | G1, GO, GI |
| l-u-1 | Lower/upper directory pages | G1, G3, G6, G4, G7, G9 |
| eu-1 | Directory-entry pages in upper directories | G1, G9 |
| u-1 | Upper directory pages | G1, G4, G7, G9 |
| u-2 | Merged upper directory pages | G1, GU |
| s-l-1 | Same/lower directory pages | G1, G2, G5, G8, G3, G6 |
| o-i-e-1 | All kinds of pages | G1, G2, G3, G4, G5, G6, G7, G8, G9 |
| o-i-e-2 | All merged pages | G1, GO, GI, GD |

considered when using SVM as the classifier. But even though SVM is good at high-dimensional classification problems, for the following reasons we limit the number of feature words (plain-text-based features) and the number of feature subset combinations:

1. Many researchers have noted the need for feature selection to make possible the use of conventional learning methods, to improve generalization accuracy, and to avoid over-fitting when using SVM. In addition, for reasons of both efficiency and efficacy, feature selection is often used when applying machine learning methods to text categorization [90], Dumais [19], and Craven [46].

2. In our research, the experiment results on different feature sets show that a larger feature set does not always perform better than a smaller one for which all the features are included in the larger feature set. For example, the feature set "u-1" contains fewer features than feature set "l-u-1" and all the features in "u-1" are in "l-u-1", but "u-1" performs better than "l-u-1".

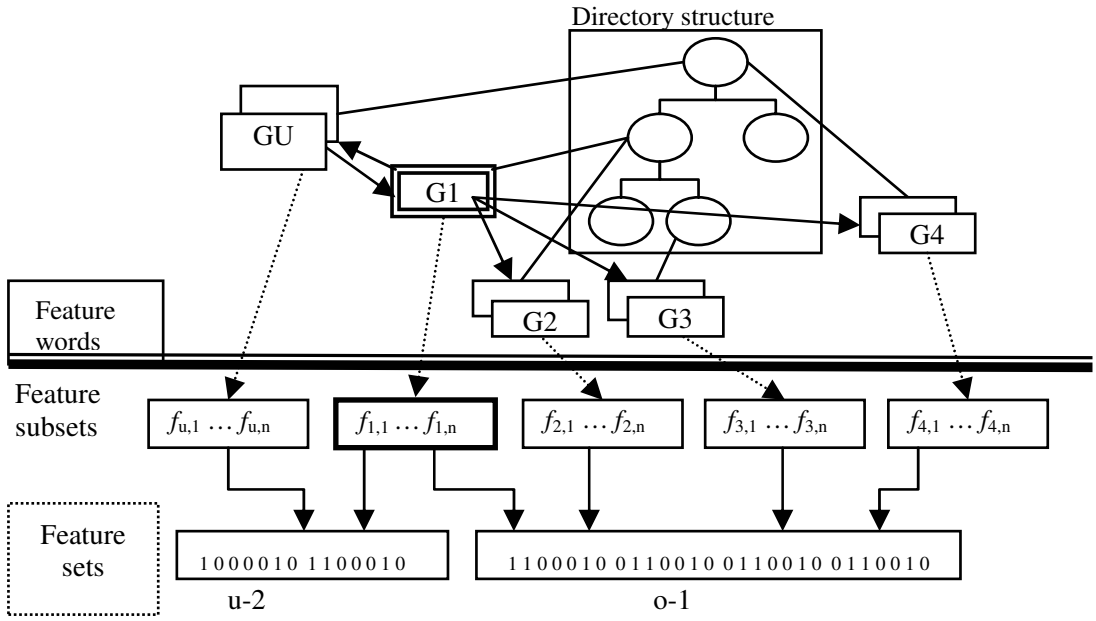


Figure 5.3: Feature subsets and feature sets.

3. In our research, feature subsets on various surrounding pages are concatenated. For example, the feature set o-i-e-1_tag was composed by concatenating feature subsets for nine times. If we restrict the number of plain-text-based features to 2,000 and use all the tagged-text-based features (around 1,000 feature words), the total number of features will be more than 27,000. If we don't restrict the number of features, there will be more than 30,000 plain-text-based feature words and the total number of features for o-i-e-1 will be at least 270,000. In this case, both the over-fitting and the efficiency of the classification would be big problems.

Therefore, both for improving the efficiency of the classification and avoiding the over-fitting of the features, it is necessary to restrict the number of the feature words and/or to test different combinations of feature subsets.

### 5.4.4 Feature Values

For both plain-text-based features and tagged-text-based features we use binary and real values. A binary value represents the presence/absence of the feature word

("1" represents presence and "0" represents absence). A real value represents the normalized page frequency of the feature word in the corresponding pages.

Let $F$ be a feature subset, $N_F$ be the number of feature words of $F$, $F_i$ be a feature word of $F$ ($1 \leq i \leq N_F$), $D$ be a set of the pages containing feature word $F_i$, $N_D$ be the total number of pages in $D$, $D_j$ be one of the pages in $D$ ($1 \leq j \leq N_D$). Then the **feature value** of feature word $F_i$ is calculated as follows:

1. $V(D_j, F_i)$ is given a binary value, 1 if $F_i$ is present in $D_j$, 0 otherwise. It is defined as:

$$V(D_j, F_i) = \begin{cases} 1 & : \quad Pres(D_j, F_i) = .T. \\ 0 & : \quad Pres(D_j, F_i) = .F. \end{cases}$$

   where $Pres(D_j, F_i)$ is whether $F_i$ is present in page $D_j$ or not.

2. The binary value of $F_i$ for page set $D$ is 1 if $F_i$ is present in any $D_j$, 0 otherwise. It is defined as:

$$V_B(F_i) = \begin{cases} 1 & : \quad \exists D_j, V(D_j, F_i) = 1 \\ 0 & : \quad \forall D_j, V(D_j, F_i) = 0 \end{cases}$$

3. The real value of $F_i$ is obtained by averaging $V(D_j, F_i)$ over $D$:

$$V_R(F_i) = \frac{\sum_{j=1}^{N_D} V(D_j, F_i)}{N_D}$$

Note that the binary and the real feature values are the same for the baseline based on the definition.

## 5.5 Experiments

### 5.5.1 Experiment Step

We used SVM$^{light}$ as a tool in the accurate classification. For all the experiments, 5-fold cross validation testing was adopted. Plain-text-based features in target pages only were used as the baseline.

We first experimented with a setting of plain-text-based features, binary feature values, and *c-j-option*-based tuning for each of the reasonable feature subset combinations. Of all the combinations of feature subsets, six with relatively high recall or precision were selected: i-1, i-e-1, eu-1, u-1, l-u-1, and o-i-e-1. With the baseline, there were seven combinations of feature subsets used for the experiments hereafter.

We next used the tagged-text-based features in addition to the plain-text-based features. All seven of the combinations of feature subsets with/without the tagged-text-based features and with the binary/real feature values were examined with the *c-j-option*-based tuning. The experiment runs with the tagged-text-based features and the real feature values are indicated by the suffixes "_tag" and "_real".

Consequently, the *offset*-based tuning was applied to the two best-performing feature sets: o-i-e-1_tag_real and u-1_tag.

Finally, to evaluate the applicability of the proposed features to homepage categories other than researchers, we did experiments on the Web->KB data set using the same feature sets used for classifying researchers' homepages.

### 5.5.2 Sample Data

For the experiments on the accurate classification, we assessed 20,846 pages (1%) randomly selected from the output of the rough filtering, thereby obtaining 481 positive pages and 20,366 negative pages. Together with 425 positive sample pages used for the rough filtering (1 page overlapped with 481 pages), the sample data set for experiments of the accurate classification contained 906 positive pages and 20,366 negative pages, 21,272 in total. Unless otherwise specified, all the experiment results below are based on these 21,272 sample pages if without additional explanation.

### 5.5.3 Experiment Results of Feature Subsets on Surrounding Pages

The experiment results of feature subsets on surrounding pages are shown in Figures 5.4-5.6. Note that each curve is drawn by connecting the upper-right-most plots of the experiment results with each feature set. In each figure, the $x$-axis is recall and

the $y$-axis is precision. The baseline and o-i-e-1 results are shown in each figure for comparison.

Figure 5.4 shows a comparison between the baseline and six typical combinations of feature subsets. It shows that feature sets including feature subsets from pages in the upper directories (e.g., u-1, eu-1) tend to perform better than feature subsets from the same and lower directories. Other experiment results (not shown) on 40 feature sets showed that all 40 outperformed the baseline in the high-recall region and that 9 outperformed the baseline in the high-precision region.

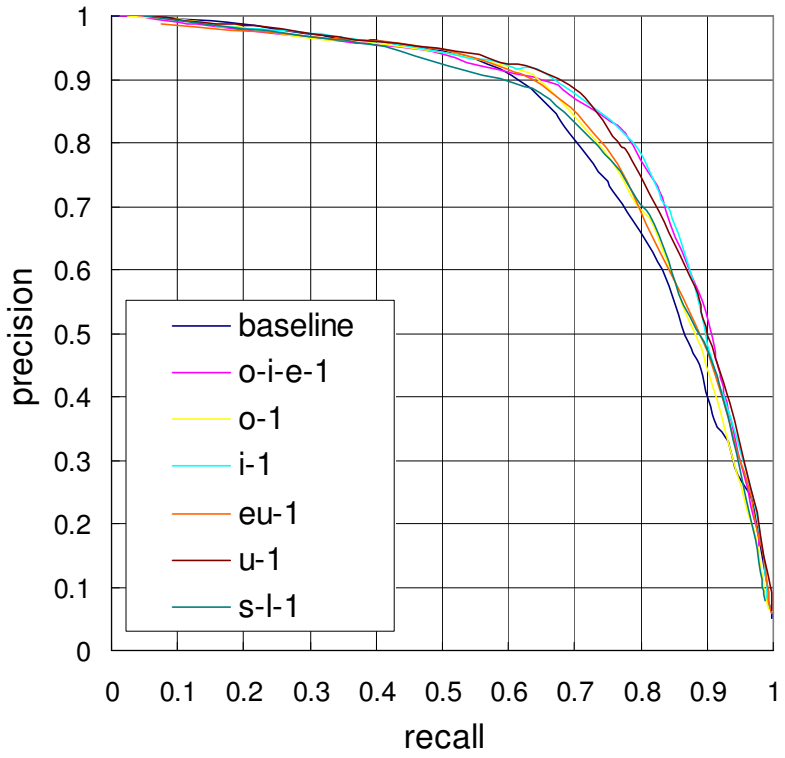


Figure 5.4: Effectiveness of feature subsets on surrounding pages.

### 5.5.4 Experiment Results of Tagged-text-based Features

Figure 5.5 shows that the tagged-text-based features are consistently effective although the degree of effectiveness depends on the feature sets.

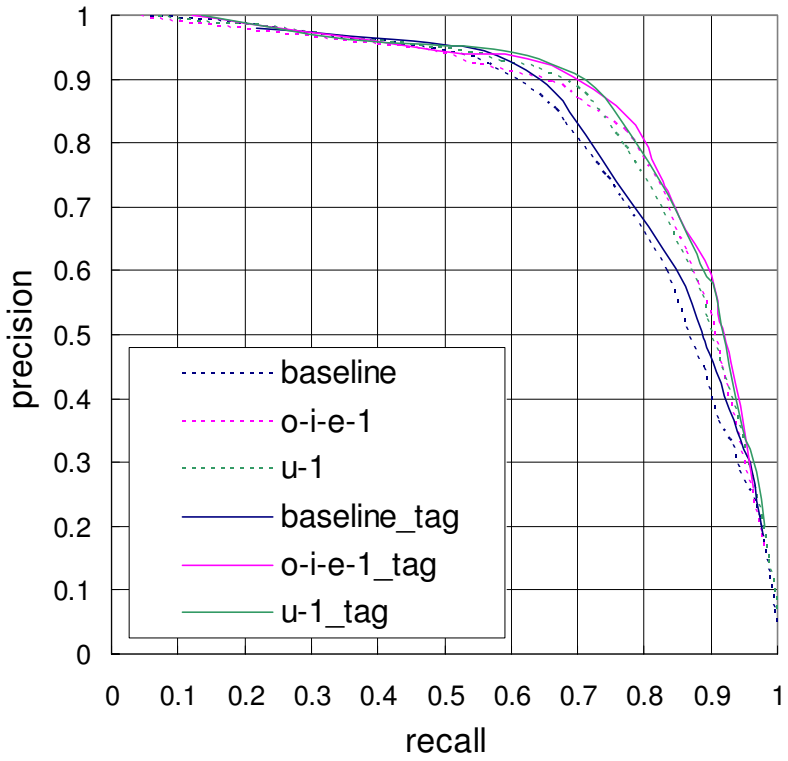


Figure 5.5: Effectiveness of tagged_text_based features.

### 5.5.5 Experiment Results of Real Values

Figure 5.6 shows the differences caused by the feature values, binary or real. The effects are unclear and in most cases are almost negligible.

### 5.5.6 Comparison of the Results of Two Tuning Methods

The experiment results obtained with *c-j-option*-based tuning are compared in Figure 5.7 with those obtained with *offset*-based tuning. The figure shows that *offset*-based tuning is inferior to *c-j-option*-based tuning in all performance ranges.

The overall experiment results show that the feature set o-i-e-1_tag_real performs the best for both the precision-assured and recall-assured classifiers. The feature set u-1_tag does not perform as well as o-i-e-1_tag_real in terms of recall, but the difference is small despite the much simpler feature composition of u-1_tag.

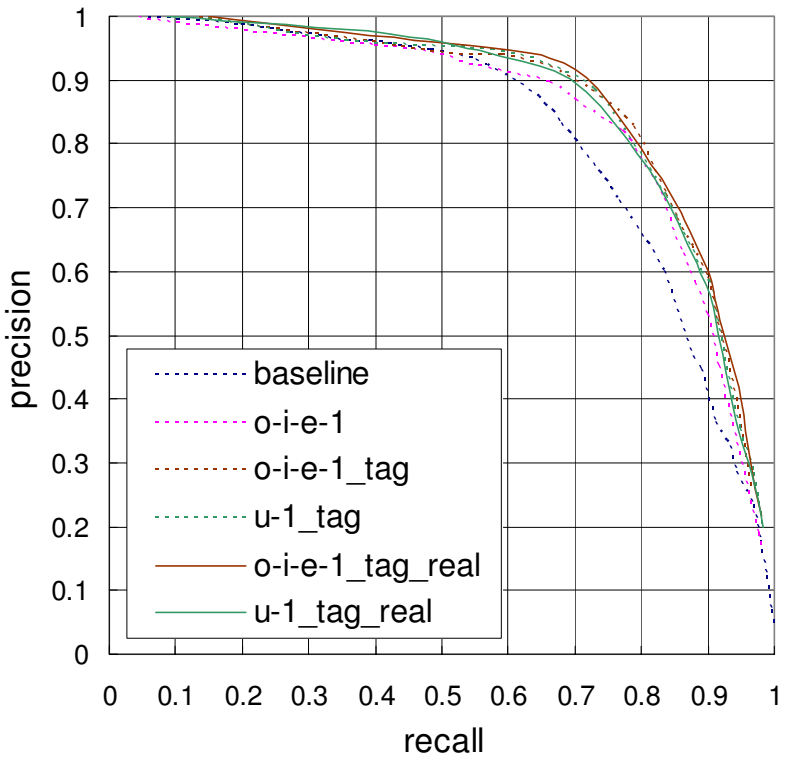


Figure 5.6: Effectiveness of real values.

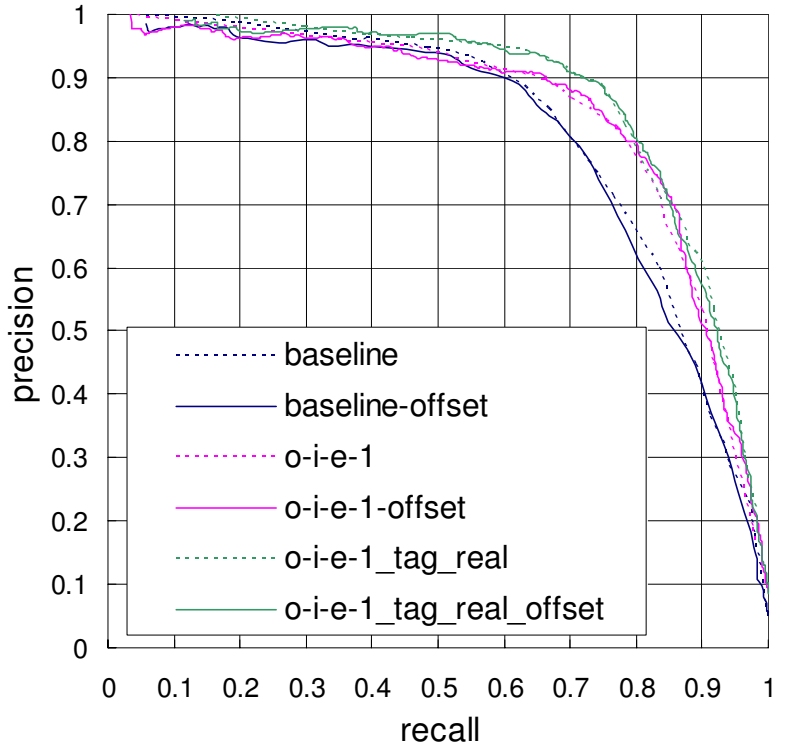


Figure 5.7: Effectiveness of *c-j-option*-based tuning over *offset*-based tuning.

### 5.5.7 Experiment on the Web->KB Data Set

To investigate the performance of the classifiers with the proposed feature sets in more general cases, we experimented with our method on the **Web->KB** data set[4], a test collection commonly used for the web page classification task.

The Web->KB data set contains 8,282 web pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base (Web->KB) project of the CMU text learning group. The 8,282 pages were manually classified into seven categories: **student** (1641), **faculty** (1124), **staff** (137), **department** (182), **course** (930), **project** (504), and **other** (3764). The "other" category is a catch-all category to which documents that do not belong in any of the defined categories of interest are assigned. Only four categories, including student, faculty, course, and project, were used in the experiments because the number of pages in the staff and department categories are too small. For each category, the data set contains pages from four universities: Cornell (867), Texas (827), Washington (1205), and Wisconsin (1263). There are also 4,120 miscellaneous pages collected from other universities. All the experiments used *leave-one-university-out* cross-validation to conduct training and testing. In the 4-fold validation, miscellaneous pages were always used as training samples.

In related experiments, for each training-testing data pair the feature words were extracted by *rainbow* software[5] with the same options as the Web->KB Project for tokenizing the data[6]. For plain-text-based features, top 2,000 words were automatically selected based on MI from the content of the pages by specifying the options of *rainbow*. For tagged-text-based features, all the *text* segments up to 4 words that match the same patterns given in subsection 5.4.2 were extracted from each page, tokenized by *rainbow* software, and used (around 600 words for each training-testing data pair). The real feature values were calculated with the same method explained in subsection 5.4.4.

We applied to the Web->KB data set the baseline and several well-performing

[4]http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

[5]http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/gentle_intro.html.

[6]http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

classifiers with the same feature sets used in the experiments described earlier in this section (i.e. o-i-e-1, u-1_tag, o-i-e-1_tag, and o-i-e-1_tag_real). The classifiers were tuned with $c$ and $j$ options and the highest F-measures for each category are listed in Table 5.4. The precision, recall, and F-measure were calculated as follows:

$$Precision = \frac{\#correct\ positive\ predictions}{\#positive\ predictions}$$

$$recall = \frac{\#correct\ positive\ predictions}{\#positive\ samples}$$

$$F - measure = \frac{2 * Precion * Recall}{Precision + Recall}$$

Table 5.4: Classification results of the Web->KB data set

| Method | course | faculty | project | student |
|---|---|---|---|---|
| baseline | 68.41 | 76.01 | 39.62 | 74.95 |
| o-i-e-1 | 76.97 | 78.27 | 53.30 | 72.53 |
| u-1_tag | 75.13 | 77.37 | 41.49 | 71.51 |
| o-i-e-1_tag | 77.82 | 79.60 | 59.49 | 74.53 |
| o-i-e-1_tag_real | 77.09 | 79.35 | 57.24 | 75.04 |

## 5.6 Considerations

### 5.6.1 Effectiveness of Web-based Features

The experiment results show that the plain-text-based feature subsets extracted from the surrounding pages in the upper directories contribute most to improving recall. This may indicate that such pages provide contextual information that is lacking in the target pages themselves (e.g., organization names and research fields). The results also shown that the surrounding pages in the same and the lower directories contribute fairly well to improving recall, though not as much as we had expected. The reason is probably that the fraction of homepages whose actual information is contained only in their component pages is not a large one. Such pages cannot be collected, however, without using features from their surrounding pages any way.

Conversely, all of the plain-text-based feature subsets from the surrounding pages have little or rather negative effects on the precision. This may indicate that the surrounding pages tend to contain confusing information that increase the noise.

In addition, the tagged-text-based features both from the target pages and the surrounding pages were found to be generally effective for improving recall, although no clear effects on precision are observed.

An interesting point to note is that when the feature subsets from surrounding pages are used together with the tagged-text-based features, they increase not only to the recall but also the precision. This may indicate that the confusing information from the surrounding pages is suppressed by the tagged-text-based features, enabling their useful information to be exploited.

### 5.6.2 Effectiveness of the Feature Sets

The best F-measures obtained when the well-performing classifiers and the baseline were used with the NW100G-01 sample data are listed in Table 5.5. To know the evident effectiveness of the proposed feature sets, we calculated confidence intervals for precision and recall based on each run result of five experiment runs of o-i-e-1_tag_real where the highest F-measure is achieved. In the results listed in Table 5.6, the F-measure is calculated by corresponding precision and recall. The results show that even in the strictest case—i.e. the lower CI of F-measure of o-i-e-1_tag_real (79.27%)—it yields an F-measure much higher than the best F-measure of the baseline (74.82%). We therefore concluded that the proposed features are effective in improving both the precision in high-recall area and the recall in high-precision area even though the best-performing recall-assured classifier and precision-assured classifier use different feature sets—respectively, i.e. o-i-e-1_tag_real and u-1_tag.

To show how much the proposed features improve the performances of the classifiers used for recall-assured and/or precision-assured classifiers, the experiment performances of the classifiers with the *c-j-option-based* tuning method are listed in Table 5.7, where each of the listed values is the average for five experiment runs.

The results of T-tests (at 95% confidence level) between each pair of baseline and candidate classifiers (o-i-e-1_tag_real and u-1_tag) based on each run result of

Table 5.5: Best F-measures of corresponding classifiers

| Classifier | o-i-e-1_tag_real | u-1_tag | baseline |
|---|---|---|---|
| precision | 88.65% | 88.58% | 83.26% |
| recall | 74.72% | 72.73% | 67.93% |
| F-measure | 81.09% | 79.88% | 74.82% |

Table 5.6: 5% confidence intervals of precision and recall at the best F-measure with the best-performing classifier

| o-i-e-1_tag_real | Performance | $\alpha$ | Lower CI | Upper CI | CI width |
|---|---|---|---|---|---|
| precision | 88.65% | 0.05 | 86.59% | 90.41% | 3.82% |
| recall | 74.72% | 0.05 | 71.82% | 78.45% | 6.63% |
| F-measure | 81.09% | 0.05 | 79.27% | 83.18% | 3.92% |

five experiment runs are listed in Table 5.8. The result shows that with regard to the precision at 95% recall, the precisions of each pair of classifiers are significantly different, while with regard to the recall at 99% precision, the recall of u-1_tag are significantly different from the recalls with the other two classifiers but the recall of o-i-e-1_tag_real is not significantly different from the recall of the baseline. We concluded that the feature subsets and feature sets we proposed are evidently effective in improving precisions in high-recall area, while the feature subsets in upper directories are more effective in improving recalls in high-precision area than

Table 5.7: Performances of recall-assured and precision-assured classifiers with two different tuning methods

| Classifier | Recall at 99% precision | Precision at 95% recall |
|---|---|---|
| o-i-e-1_tag_real | 20.86% | 41.33% |
| u-1_tag | 23.18% | 33.22% |
| baseline | 17.98% | 26.49% |

Table 5.8: T-test results on performance of the classifiers

| **Performance** | **Recall at 99% precision** | | | **Precision at 95% recall** | | |
|---|---|---|---|---|---|---|
| Method | o-i-e-1_tag_real | u-1_tag | baseline | o-i-e-1_tag_real | u-1_tag | baseline |
| o-i-e-1_tag_real | | 0.0025 | 0.0986 | | 0.0000 | 0.0071 |
| u-1_tag | | | 0.0180 | | | 0.0000 |
| baseline | | | | | | |

other kinds of feature subsets.

As mentioned in subsection 5.5.7, we applied the well-performing classifiers with the same feature sets used for classifying researchers' homepages to four category pages in the Web->KB data set. To evaluate the applicability of the proposed method to web pages of other categories in other languages, we compared our experiment results with those reported for seven related studies that also used the Web->KB data set.

The information utilized for web page classification in each study and the experiment results obtained on the Web->KB data set are listed together with the corresponding information for other seven related studies in Table 5.9 and Table 5.10. FOIL(Linked Names) is a first order inductive learner algorithm for web page classification that uses features extracted from the hypertext [47], while $k$NN(Tagged Words) classifies web pages by using the $k$-nearest neighbor algorithm and distinguishing the content in the pages and from the content in linked pages [47]. The SVM(TA) method [72] classifies web pages by using both the text and context feature sets in the pages, such as X (text only), T (title), A (anchor). SVM-FST is a web classification method exploring URLs via a two-phase pipeline of word segmentation/expansion and classification [34]. ME-w is a web page classification method that uses maximum-entropy(ME)-based learning on plain text and/or URL text [74]. The SVM-iWUM($\alpha = 1$) (the iterative web unit mining) method classifies web pages according to the information on subgraphs of webpages that can be utilized from web site structure [48]. The GE-CKO(FC5) method uses the composite kernels to

optimize the linear combination of kernels for web page classification [73].

Table 5.9: Information utilized by previous methods

| Method | Information utilized |
|---|---|
| Our method | plain text, tag text, local in-/out-link, directory entry page, URL |
| FOIL(Linked Name) [47] | local linked or anchor text |
| $k$NN(Tagged Words) [47] | plain text and text from linked pages distinguished |
| SVM(TA) [72] | plain text, title, anchor text |
| SVM-FST(XATU) [34] | plain text, anchor text, title, and URL text |
| ME-w(TU) [74] | plain text and URL text |
| SVM-iWUM($\alpha = 1$) [48] | URL, local link, directory entry page, title, number of links, in-link anchor |
| GE-CKO(FC5) [73] | plain text, in-/out-link, title, anchor text |

Table 5.10 show that our method outperformed all seven of the previous methods in terms of the macro-averaged F-measure of all the four categories (Macro(4)) and was a little inferior to only one of the seven previous methods in terms of the macro-averaged F-measure for the course, faculty, and student categories (Macro(3)). Our method outperformed ten out of twelve on per-category basis (F-measures of the individual categories are not available for 4 of the previous studies). We can therefore conclude that the proposed feature sets perform fairly well and are applicable not only to researchers' homepages in Japanese but also to other categories in other language.

Note that we cannot discuss the significance of the performance on the Web->KB data set in comparison to previous studies and have not applied the accurate classification (the three-way classifier) to the Web->KB data set because there is no previous work for comparison. Because the feature sets work quite well, however, we think that the proposed method is applicable.

Table 5.10: Classification performances of previous methods

| Method | course | faculty | project | student | Macro(4)* | Macro(3)** |
|---|---|---|---|---|---|---|
| **o-i-e-1_tag** | 77.8 | 79.6 | 59.5 | 74.5 | 72.9 | 77.3 |
| **o-i-e-1_tag_real** | 77.1 | 79.4 | 57.2 | 75.0 | 72.2 | 77.2 |
| FOIL(Linked Names) | | | | | 62.9 | |
| $k$NN(Tagged Words) | | | | | 59.1 | |
| SVM(TA) | 68.2 | 65.9 | 32.5 | 73.0 | 59.9 | 69.0 |
| SVM-FST(XATU) | 60.9 | 40.9 | 66.5 | 25.3 | 48.4 | 42.4 |
| ME-w(TU) | | | | | 62.7 | |
| SVM-iWUM($\alpha = 1$) | 54.7 | 87.6 | 17.1 | 95.8 | 63.8 | 79.4 |
| GE-CKO(FC5) | | | | | | 76.5 |

Note: All the performances are F-measures. *Macro(4) is the average F-measure for all four categories; **Macro(3) is the average F-measure for all categories other than category project.

### 5.6.3 Effectiveness of the Tuning Methods

The experiment results in Figure 5.7 show that for both a given recall and a given precision, the *offset*-based tuning is considerably inferior to the *c-j-option*-based tuning. It is understood that changing the *offset* is equivalent to a parallel translation of the separating hyperplane and the degree of freedom is one and that adjusting $c$ and $j$ optimizes both the normal vector and the *offset* of the hyperplane and the degree of freedom is then equal to the order of the feature space. Therefore the latter tuning method can adapt better to skewed distributions of the positive data over the negative data and can consequently yield higher precision. Because the negative data is widely distributed, however, its skew over the positive data has little influence on the recall.

We conducted T-tests at 95% confidence level between the classifiers using two tuning methods at a specific performance requirement for the baseline, u-1_tag, and o-i-e-1_tag_real. The performances obtained by tuning *offset* for the classifiers are

listed in Table 5.11 and the upper part of the table is the same as Table 5.7. Each value in this Table is the average for five experiment runs.

Table 5.11: Performances obtained with two tuning methods

| **Classifier** | **Recall at 99% precision** | **Precision at 95% recall** |
|---|---|---|
| o-i-e-1_tag_real (O) | 20.86% | 41.33% |
| u-1_tag (U) | 23.18% | 33.22% |
| baseline (B) | 17.98% | 26.49% |
| o-i-e-1_tag_real-*offset* (O-*off*) | 13.25% | 38.49% |
| u-1_tag-*offset* (U-*off*) | 10.49% | 36.48% |
| baseline-*offset* (B-*off*) | 6.22% | 25.69% |

The performances in Table 5.11 show that for the precision-assured classifier the tuning method is the dominant factor determine performance: all the performances obtained with the *c-j-option-based* tuning method are better than those obtained with the *offset-based* tuning method. For the recall-assured classifier, in contrast, the feature set is the dominant factor: o-i-e-1_tag_real performs best and the baseline performs worst. We conducted T-tests (at 95% confidence interval) on each pair of the classifiers for both precision-assured and recall-assured classifiers based on each run result of five experiment runs. The results are listed in Tables 5.12 and 5.13, where the methods are sorted in descending order of the performance and the italic font is for $p>0.05$.

The results show that at 99% precision, all the recalls obtained by the *c-j-option-based* tuning method are significantly different from those obtained by the *offset-based* tuning method, while at 95% recall, with both tuning methods not only is the precision obtained by o-i-e-1_tag_real significantly different from those obtained by u-1_tag and baseline but also that obtained by u-1_tag is significantly different from that obtained by the baseline. We therefore concluded that the *c-j-option-based* tuning method is consistently effective in improving the recall of precision-assured classifiers and that the *c-j-option-based* tuning method is also effective for recall-

assured classifiers when the feature word number is large.

Table 5.12: T-test results on recalls at 99% precision

| **Method** | U | O | B | O-f | U-f | B-f |
|---|---|---|---|---|---|---|
| **recall** | 23.18% | 20.86% | 17.98% | 13.25% | 10.49% | 6.22% |
| U | | 0.0025 | 0.0180 | 0.0000 | 0.0000 | 0.0000 |
| O | | | *0.0986* | 0.0000 | 0.0000 | 0.0000 |
| B | | | | 0.0381 | 0.0060 | 0.0002 |
| O-f | | | | | *0.0782* | 0.0002 |
| U-f | | | | | | 0.0108 |
| B-f | | | | | | |

Table 5.13: T-test results on precisions at 95% recall

| **Method** | O | O-f | U-f | U | B | B-f |
|---|---|---|---|---|---|---|
| **precision** | 41.33% | 38.49% | 36.48% | 33.22% | 26.49% | 25.69% |
| O | | 0.0027 | 0.0000 | 0.0000 | 0.0071 | 0.0000 |
| O-f | | | 0.0282 | 0.0000 | 0.0000 | 0.0000 |
| U-f | | | | 0.0063 | 0.0000 | 0.0000 |
| U | | | | | 0.0000 | 0.0002 |
| B | | | | | | *0.1300* |
| B-f | | | | | | |

### 5.6.4 Reduction of Manual Assessment

To evaluate the reduction of the amount of pages requiring manual assessment (i.e., the pages classified as uncertain), we compares two compositions of the three-way classifier, one using the baseline and the other using the o-i-e-1_tag_real, for both recall-assured and precision-assured classifiers.

Table 5.14 shows, for the 100GB data set, the estimated numbers of pages in each output class for three different quality requirements. The page numbers are estimated from the output proportions of the 1% sample data collected directly from the output of the rough filtering. The reduction ratios of the pages labeled uncertain are also listed.

Table 5.14: Estimated numbers of pages in each classification output from the 100GB data set

| **Assured performance** | **baseline** | | | **o-i-e-1_tag_real** | | | **Reduction ratio** |
|---|---|---|---|---|---|---|---|
| **precision/recall** | positive | uncertain | negative | positive | uncertain | negative | |
| 99.5%/98% | 3,780 | 461,832 | 1,618,988 | 9,206 | 358,207 | 1,717,187 | 77.6% |
| 99%/95% | 8,805 | 275,517 | 1,800,278 | 10,204 | 157,829 | 1,916,567 | 57.3% |
| 98%/90% | 9,870 | 156,664 | 1,918,066 | 15,503 | 81,157 | 1,987,940 | 51.8% |

The results show that, with the proposed features and the three-way classifier composition, the amount of pages that need to be manually assessed can be reduced markedly, especially when the assured performance is relaxed.

Note when we did example analysis (described in the following subsection), we found that some positive pages were misjudged as negative and this may have had some bad effects on the SVM-based classifier. Nevertheless, we are sure that our method can reduce the amount of pages that need to be manually assessed in order to assure the quality of the homepage collection. The degree to which the amount of such pages can be reduced, however, will have to be determined in further experiments.

We have not yet had time to evaluate the accurate classification on the larger (1.36TB) data set. We will do this later.

### 5.6.5 Analysis of Classification Result Examples

We use two sets of positive samples, one from the Jname data (the 425 pages obtained by excluding the page that overlapped with the other part of the positive samples) and the other from 1% of the output of the rough filtering (481 pages). The classification results with o-i-e-1_tag_real and the baseline are listed in Tables 5.15 and Table 5.16.

Table 5.15: Classification result using o-i-e-1_tag_real as the three-way classifier (95.04% recall and 98.96% precision)

| **Samples** | **positive** | **uncertain** | **negative** | **Total** |
|---|---|---|---|---|
| positive from the rough filtering | 76 | 365 | 40 | 481 |
| positive from Jname data | 113 | 307 | 5 | 425 |
| negative | 2 | 1,223 | 19,141 | 20,366 |
| false positive | 2 | | | |
| false negative | | 45 | | |
| Total | 191 | 1,895 | 19,186 | 21,272 |

Table 5.16: Classification result using the baseline as the three-way classifier (95.11% recall and 99.11% precision)

| **Samples** | **positive** | **uncertain** | **negative** | **Total** |
|---|---|---|---|---|
| positive from the rough filtering | 64 | 382 | 35 | 481 |
| positive from Jname data | 97 | 314 | 14 | 425 |
| negative | 2 | 2,381 | 17,983 | 20,366 |
| false positive | 2 | | | |
| false negative | | 48 | | |
| Total | 163 | 3,077 | 18,032 | 21,272 |

The results show that the two sets of positive samples cannot be assumed to be the same and that o-i-e-1_tag_real can reduce more pages that need manual

assessment comparing to that the baseline can.

### Success Examples for Precision-assured Classifier

We first did success analysis for the precision-assured classifier. That is, we analyzed positive samples successfully labeled as positive with o-i-e-1_tag_real but labeled as negatived with the baseline. Figure 5.8 and 5.9 are two typical examples of success cases for the precision-assured classifier. The typical examples are the pages with limited useful contents but even with some noise information. Their failure with the baseline may be because, to make the precision very high, such obviously positive pages cannot be judged as positive confidently just with the information of word occurrences in the target pages alone.

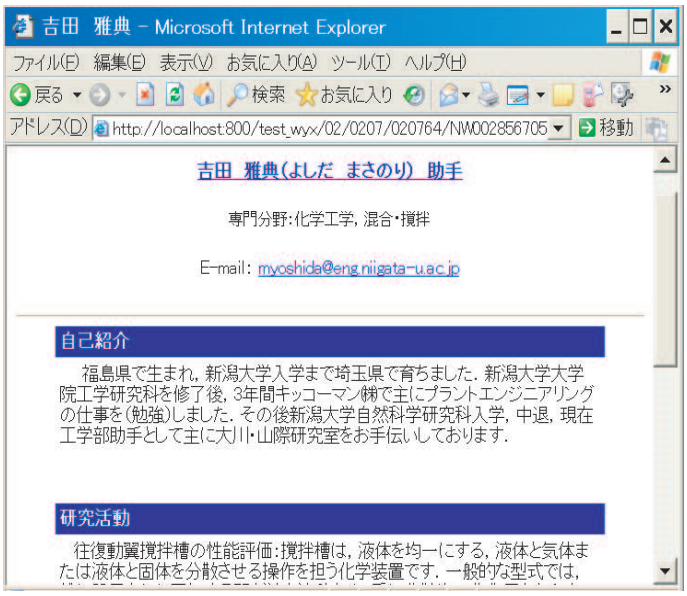


Figure 5.8: Success example 1 for precision-assured classifier.

### Failure Examples for Precision-assured Classifier

We then did failure analysis for the precision-assured classifier. That is, we analyzed negative samples labeled as positive even with o-i-e-1_tag_real. By checking the two false positive pages listed in Table 5.15, we found that both of them should have been judged as positive and that one was probably over-looked and the other was misjudged because it begins with the introduction of the professor's laboratory.

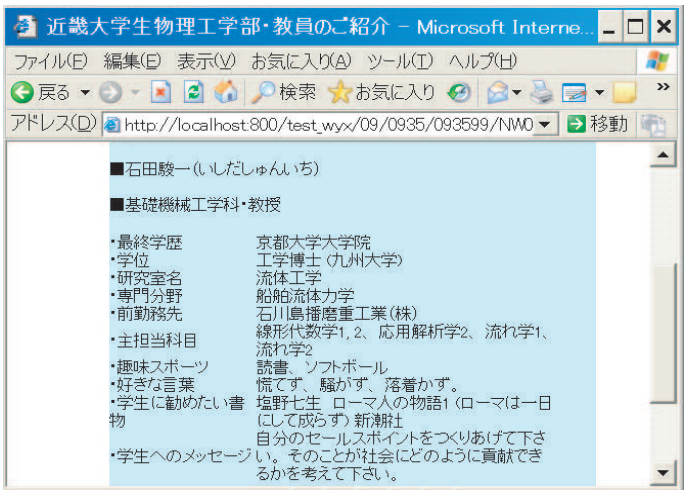


Figure 5.9: Success example 2 for precision-assured classifier.

Without the two misjudged false positive pages, the precision of the precision-assured classifier will become 100%. To find some real failure examples for the precision-assured classifier, we relaxed the precision to 97.6% and 95%. Figures 5.10, 5.11, and 5.12 show three typical failure examples for the precision-assured classifier. We found from the typical failure examples that the failures (excluding two more misjudgments) occur mainly in the three cases listed in Table 5.17.

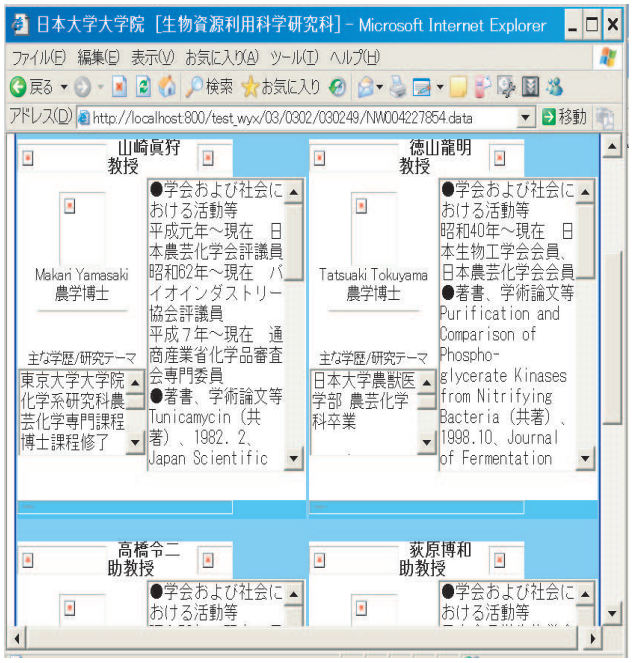


Figure 5.10: Failure example 1 for precision-assured classifier.

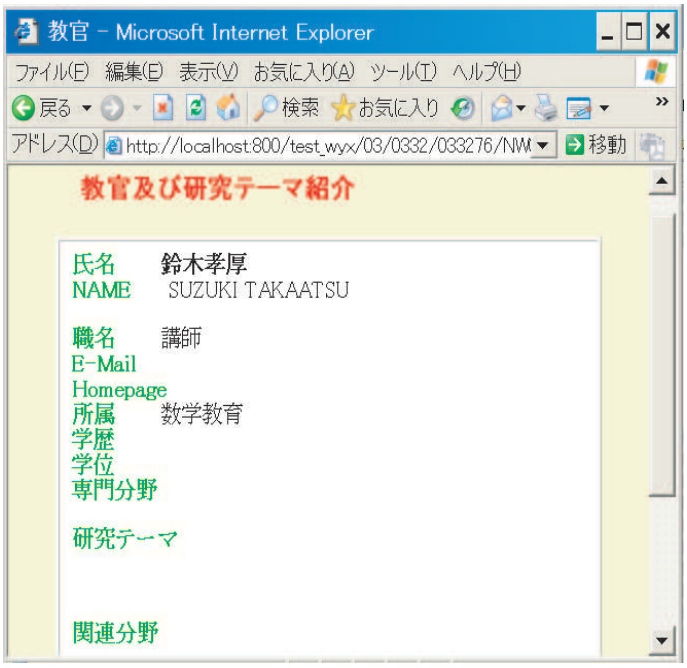


Figure 5.11: Failure example 2 for precision-assured classifier.

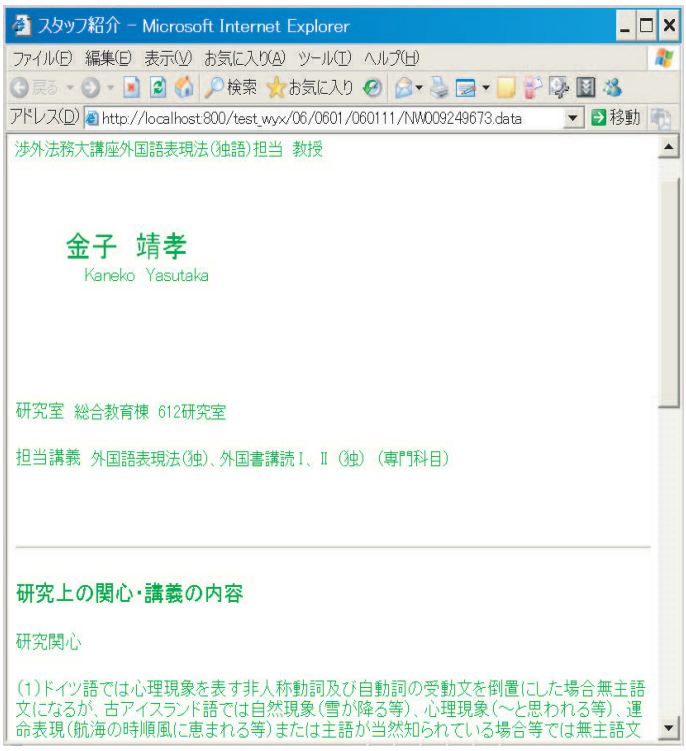


Figure 5.12: Failure example 3 for precision-assured classifier.

Table 5.17: Analysis of false positive pages

| Type | Reason |
|---|---|
| 1 | More than one researchers' introductions are in one page |
| 2 | Has the official homepage format but limited information, e.g. only the name |
| 3 | Introduction to the department, the laboratory, the course, etc. |

The analysis result shows that even with the information in surrounding pages, pages of types 1 and 3 are hard to classify correctly. For type-2 pages, the surrounding pages might have had bad effects. Pages of types 1 and 2, however, might provide useful information. Note that type-2 pages can never be found with the baseline.

### Success Examples for Recall-assured Classifier

We next did success analysis for the recall-assured classifier. That is, we analyzed negative samples successfully labeled as negative with o-i-e-1_tag_real but labeled as positive with the baseline. Figures 5.13, 5.14, and 5.15 show three typical success examples for the recall-assured classifier: project introduction pages, syllabus pages, and pages on paper introduction, recruitment of research associates, etc. These examples show that, in order to make the recall very high, such pages cannot be excluded just with the information of word occurrences in the target pages alone. That is, the information in the surrounding pages is needed for assuring a high recall.

### Failure Examples for Recall-assured Classifier

Finally, we did failure analysis for the recall-assured classifier. That is, we analyzed positive samples labeled as negative. The results show that there were 14 misjudged pages and the typical causes of the misjudgments were as follows. Even though we understand the criterion for judging researchers' homepages, it is still difficult to applying to some web pages in some cases. In addition, the judgment needs to be

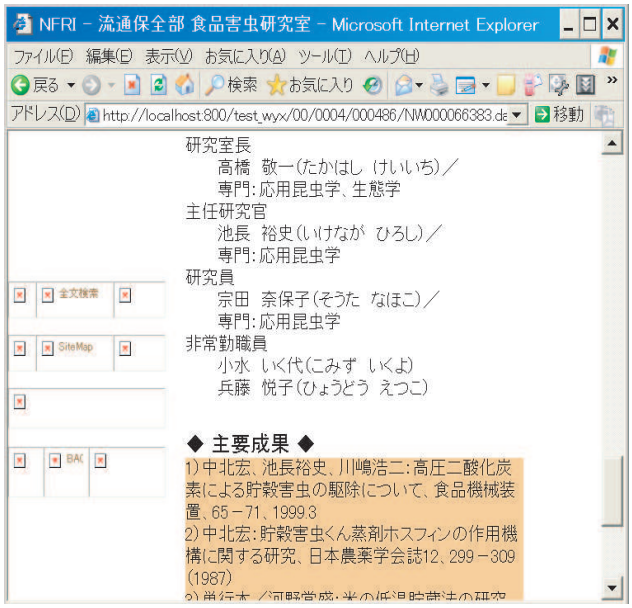


Figure 5.13: Success example 1 for recall-assured classifier.

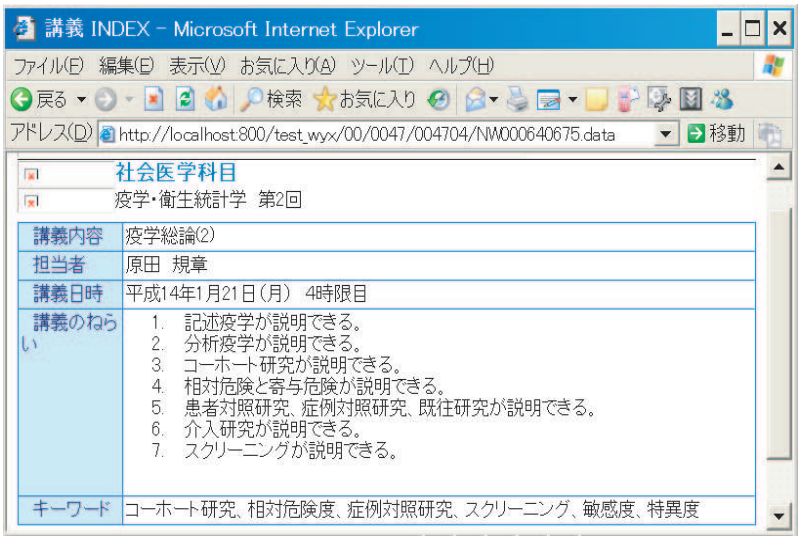


Figure 5.14: Success example 2 for recall-assured classifier.

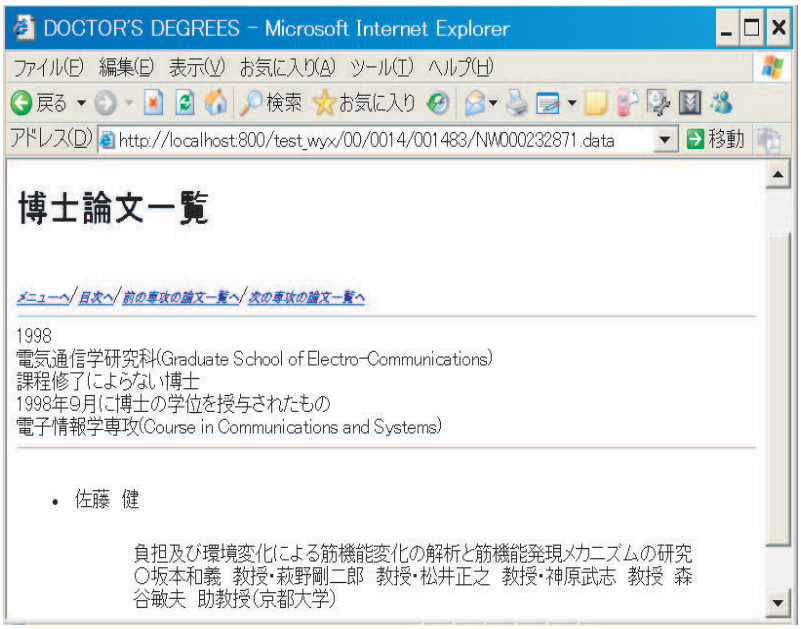


Figure 5.15: Success example 3 for recall-assured classifier.

based on the surrounding pages for which errors tend to happen. For the real false negatives, the main difficulties for the recall-assured classifier are listed in Table 5.18.

Table 5.18: Analysis of false negative pages

| Source | # | Reason |
|---|---|---|
| from Jname data | 5 | Has limited contents and uses informal words |
| from the rough filtering | 4 | The entry pages are in English and contain little related information in Japanese |
| | 12 | Too little information in entry pages and surrounding pages |
| | 10 | Too much noise and little research-related information |

Figures 5.16 and 5.17 show two typical failure examples for the recall-assured classifier. They are surely researcher's homepages, and the reason that the first example cannot be correctly classified as positive even with our method (i.e., o-i-e-1_tag_real) is that there is only limited information in the page itself and there is no position/title, theme, bibliography even in the only out-linked page (the site top

page). Furthermore, the two in-linking pages are member list pages, one in Japanese and the other in English. In the second example there are too many informal words and this may confuse the classifier even with the appropriate information in related surrounding pages.


Figure 5.16: Failure example 1 for recall-assured classifier.

The analysis result also shows that some of the positive samples from the Jname data are cleaner than others. It is obvious that the judgment was very difficult because we tried to primarily use the information in the surrounding pages at most. Therefore the assumption on that the two parts of positive samples are the same was incorrect. Without the 14 misjudged pages, the number of the positive samples from the rough filtering becomes 467 and the recall for this part of positive sample becomes 94.4%, which is decreased by only less than 0.6%. Therefore, even though the assumption does not hold, the experiment results for the recall-assured classifier still show the effectiveness of proposed method.

There is a possibility that the misjudged pages had some bad effects on the training of SVM. Some misjudgment cannot be avoided in actual applications, however, and the analysis results show the robustness of our method.

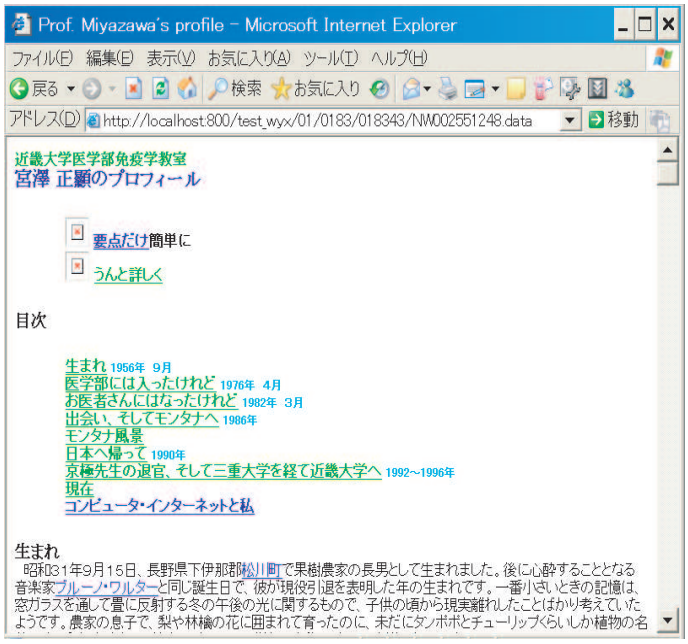


Figure 5.17: Failure example 2 for recall-assured classifier.

## 5.7 Conclusion

In this chapter we described a method for accurately classifying the candidate web pages output from the rough filtering into three classes—assured positive, assured negative, and uncertain—while assuring a required recall and precision. We use Support Vector Machine (SVM) with textual features, both plain-text-based and tagged-text-based, obtained from each page and its surrounding pages. The surrounding pages are grouped according to connection types (in-link, out-link and directory entry) and relative URL hierarchy (same, upper, or lower in the directory hierarchy), and then an independent feature subset is generated from the pages of each group. Feature subsets are concatenated conceptually to compose the feature set of a classifier. We used the *c-j-option*-based method to tune the classifiers to a specified level of performance.

The experiment results showed that the introduced features and feature sets are more effective than the baseline and that the tuning method is also effective.

When the proposed classifier is used as a recall-assured classifier and a precision-assured classifier in combination to compose the three-way classifier, the amount of pages that need manual assessment is reduced evidently, especially when the assured

performance is relaxed.

The results of the experiments in which well-performing feature sets were applied to the Web->KB data set show that the proposed feature set performs fairly well in comparison with previous methods and is applicable not only to researchers' homepages in Japanese but also to other categories of homepages in other language. The analysis of classification result examples shows the robustness of the proposed method and its effectiveness in classifying difficult pages.

In summary, the method of accurate classification is effective for achieving the required high performance and reducing the amount of manual assessment required. The proposed classifier is also estimated to be readily to other categories of homepages and to homepages in languages other than Japanese.

# Chapter 6

# System Processing Time

As discussed in Chapters 4 and 5, we have investigated the methods for the rough filtering and the accurate classification by using the manually created sample data. Taking into consideration the page group structures, the rough filtering can efficiently narrow down the amount of candidate pages with a high recall and the accurate classification can accurately classify the candidate pages into three classes—assured positive, assured negative, and uncertain—with both required high recall and high precision.

Both the methods have been shown experimentally to be effective and to be able to gather a high-quality collection of homepages if pages classified as "uncertain" are further assessed manually. The classification method has also been shown to reduce the amount of manual assessment required in order to assure a specified quality level. We can therefore conclude that we have made progress toward the objectives mentioned in Chapter 3.

This chapter discusses the computer processing cost of the system we investigated and also discusses ways to reduce it.

The computer processing cost includes the time needed for (1) page content processing, (2)feature generation, and (3) SVM running. The page content processing includes tag processing, morphological analysis, and so on, and the morphological analysis is the most time-consuming process because it includes dictionary look-up. The feature generation mainly relies on the algorithms and it can be made very fast. As for the SVM running, the classifying is very fast once the classifier model has

been learned. Therefore, (2) and (3) are so much faster than (1) that their processing time can practically be ignored. Thus only the time of page content processing is to be considered when estimating the computer processing time.

## 6.1 The Rough Filtering

The first processing step, rough filtering, can filter out the obviously irrelevant pages fast enough and it is efficient for narrowing down the huge amount of candidate pages with the required recall. Consequently, its computer processing cost is negligible.

As shown by the values in Table 4.7, the number of candidate pages is reduced to less than 23% of the pages in the corpus when we use the combination of PGM-Od@20(0,2), Ou#(1,3), I#(0,3), U#(0,3) and set the threshold score of the candidate pages to 4.

It is obvious that if the rough filtering is not used and the corpus is directly input to the next processing step, page content processing is required for all the web pages.

## 6.2 The Accurate Classification

The second processing step, accurate classification, is for accurately classifying the candidate pages with high performance into three classes: assured positive, assured negative, and uncertain. In Chapter 5, we introduced the features generated from the page contents and simple tagged-text-based contents of both the target pages and surrounding pages. Therefore the amount of pages that need to be processed will be more than that input from the output of the rough filtering.

Because the manual assessment cost is directly connected to the number of uncertain pages, that number must be kept as small as possible. Under this constraint, the computer processing cost of the classification should at the same time be reduced as much as possible.

In the following subsections, we will introduce a new structure for the recall-assured classifier, one expected to make the computer processing cost as low as

possible, and will discuss the results of experiments comparing the processing cost for this structure with that for the original one.

### 6.2.1 Cascaded Structure of the Recall-assured Classifiers

The overall experiment results in Chapter 5 show that a recall-assured classifier using the u-1_tag feature set does not perform as well as one using the o-i-e-1_tag_real feature set but the performance difference seems small when one considers the much simpler composition of the u-1_tag feature set.

A potential method to compose a recall-assured classifier with the same performance as the single best-performing classifier with less computer processing cost is to additionally use the other classifier with a little inferior performance but with much lower cost in a cascaded structure. That is, to use the relatively low cost one, u-1_tag, as the first recall-assured classifier and to use the more expensive one, o-i-e-1_tag_real, as the second recall-assured classifier, as shown in Figure 6.1.

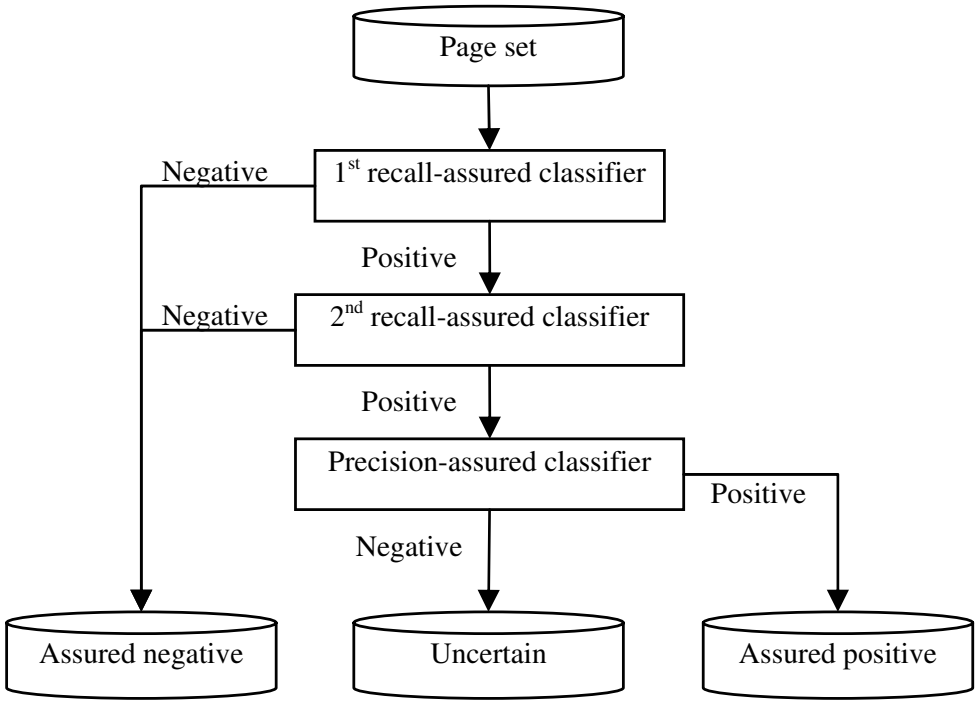


Figure 6.1: Composition of the accurate classification using a cascaded structure of recall-assured classifiers.

With this construction, to keep the same recall as the original one, the number of the false-negatives for both classifiers must be kept the same as the number of false-negative for the original one. This may increase the manual assessment cost to some degree.

### 6.2.2 Experiment Results

With the cascaded structure of the recall-assured classifiers introduced above, we did experiments for achieving at least 95% recall with various parameters ($c$ and $j$, or *offset*) of the two classifiers, u-1_tag as the first and o-i-e-1_tag_real as the second.

We first studied the performance of each classifier and then calculated their combined performances. The performances of the cascaded structure are listed in Table 6.1 and note only the performances of the combinations with 95.03% recall are listed in the table.

Table 6.1: Performance at 95.03% recall of cascaded structure of the recall-assured classifiers

| u-1_tag | | | | | | o-i-e-1_tag_real | | | | | | | **Cascaded** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Id | $c$ | $j$ | Rec. | Prec. | Op | Id | $c$ | $j$ | *offset* | Rec. | Prec. | Op | Prec. | Op |
| | | | | | | (a) | 0.00019 | 43 | 0 | 95.03 | 41.28 | 9.81 | 41.28 | **9.81** |
| (1) | 0.00005 | 90 | 99.12 | 13.97 | 30.23 | (b) | 0.00021 | 56 | 0 | 95.14 | 40.18 | 10.10 | 40.12 | 10.09 |
| (2) | 0.00008 | 60 | 98.01 | 22.40 | 18.67 | (d) | 0.00020 | 53 | 0 | 95.36 | 39.78 | 10.22 | 39.88 | 10.15 |
| (3) | 0.00009 | 50 | 97.13 | 27.31 | 15.17 | (g) | 0.00019 | 59 | 0 | 95.69 | 38.12 | 10.71 | 39.07 | 10.36 |
| (4) | 0.00012 | 60 | 97.02 | 27.75 | 14.92 | (f) | 0.00019 | 58 | 0 | 95.58 | 38.19 | 10.68 | 39.01 | 10.38 |
| (1) | 0.00005 | 90 | 99.12 | 13.97 | 30.23 | (k) | 0.005 | 2 | 1.03 | 95.58 | 35.20 | 11.56 | 38.03 | 10.64 |
| (2) | 0.00008 | 60 | 98.01 | 22.40 | 18.67 | (n) | 0.005 | 2 | 1.06 | 96.02 | 33.12 | 12.35 | 39.75 | 10.18 |
| (3) | 0.00009 | 50 | 97.13 | 27.31 | 15.17 | (p) | 0.005 | 2 | 1.08 | 96.24 | 31.98 | 12.82 | 41.04 | **9.86** |

Note: Op is the amount of pages labeled positive by the classifier as a percentage of the pages input to the classifier. Prec. and Rec. are also percentages.

The results show that even when the recall is kept at the required high level (95.03%), the proportional output of the recall-assured classifier with the cascaded structure can be almost the same as that of the original one (comparing the best

values, 9.86% for the cascaded structure and 9.81% for the original structure). The difference is only 0.05% of the input.

### 6.2.3 Reduction of the Processing Time

We can estimate the computer processing cost of the accurate classification using a single recall-assured classifier and of the accurate classification using a cascaded structure of recall-assured classifiers. Since not only the target pages but also the surrounding pages need to be processed, we estimated the proportions of the surrounding pages both for the 100GB data set and a 1.36TB data set with the expectation that the reduction of the processing cost can be more effective for a larger data set. All the estimates described here were done for both the data sets. The estimates of the proportion of the processed pages are listed in Table 6.2.

Table 6.2: Estimate amounts of processed pages as percentages of the pages in the corpus

| **Data set** | Input pages* | Input pages and their surrounding pages that need to be processed for o-i-e-1_tag_real | Input pages and their surrounding pages that need to be processed for u-1_tag |
|---|---|---|---|
| 100GB | 22.93% | 47.34% | 23.68% |
| 1.36TB | 14.74% | 29.43% | 15.17% |

*Input pages means the input from the rough filtering to the accurate classification.

The estimates of the surrounding pages show that for the u-1_tag classifier about 1% more of both the 100GB and 1.36TB data sets need computer processing, while for the o-i-e-1_tag_real classifier about 15% more of the 100GB data set and about 24% more of the 1.36TB data set need computer processing. For both 100GB and 1.36TB data sets the increase is almost the same as the input from the rough filtering to the accurate classification.

In the original structure, about 47% of 100GB data set and about 30% of 1.36TB data set need computer processing, while in the cascaded structure, since u-1_tag is used as the first classifier and only the pages output from u-1_tag (15.17% of the

input to it) need further computer processing by o-i-e-1_tag_real (approximately the same amount as the input pages), the percentage of pages in the 100GB data set that needs computer processing is $23\% + 1\% + 23\% * 15.17\% * 100\% = 27.5\%$, and the percentage of pages in the 1.36TB data set that needs computer processing is $15\% + 1\% + 15\% * 15.17\% * 100\% = 18.3\%$.

As the 100GB and 1.36TB data sets are respectively 11,038,720-page and 95,870,352-page data sets, the amount of pages that need computer processing is calculated by multiplying proportion by the number of pages in each corpus.

In addition, since the features for o-i-e-1_tag_real are processed at the recall-assured classifiers, no extra computer processing is needed at the precision-assured classifier. The amounts of pages that need computer processing in the accurate classification using a single recall-assured classifier and the abovementioned cascaded structure are listed in Table 6.3.

Table 6.3: Amount of pages that need to be processed in the accurate classification

| **Type of recall-assured classifier** | **100GB data set** | | **1.36TB data set** | |
|---|---|---|---|---|
| | **Percentage** | **Number** | **Percentage** | **Number** |
| Without the rough filtering | 100.0% | 11,038,720 | 100.0% | 95,870,352 |
| Original | 47.0% | 5,188,200 | 30.0% | 28,761,000 |
| Cascaded* | 27.5% | 3,035,500 | 18.3% | 17,544,000 |

*Combination of (3) and (p) in Table 6.1.

It is shown in Table 6.3 that to obtain at least 95% recall when using the cascaded structure instead of the single classifier, the percentage of processed pages can be reduced from 47% to 27.5% (by 19.5 percentage points) for the 100GB data set, and from 30% to 18.3% (by 11.7 percentage points) for the 1.36TB data set. For both data sets, the processed pages are reduced by about 40%. Since the amount of web pages is enormous, a 40% reduction is quite significant.

We then examined the amount of pages that need to be manually assessed when the cascaded structure of the recall-assured classifiers is used. The pages that require manual assessment are those labeled positive by the recall-assured classifier but labeled negative by the precision-assured classifier. Since less than 1% of the

pages are labeled positive by the precision-assured classifier, almost all the pages labeled positive by the recall-assured classifier need manual assessment. As shown in Table 6.1, there are more false-negatives that should be taken into account when the cascaded structure is used and this results in the proportion of output pages increasing a little (0.05% of the input pages to the classifier) even for the best-performing combination of two recall-assured classifiers.

The increased amount of pages that need manual assessment and the decreased amount of pages that need computer processing can be compared, for both the 100GB data set and the 1.36TB data set, in Table 6.4. It shows that with the use of the cascaded structure of the recall-assured classifiers, the ratio of the increase of the pages that need manual assessment to the reduction of the pages that need computer processing is less than 1:1500.

Table 6.4: Comparison of reduced computer processing pages and increased manual assessment pages using the cascaded structure of recall-assured classifiers instead of the original classifier

| **Difference** | **100GB data set** | | **1.36TB data set** | |
|---|---|---|---|---|
| | **Percentage** | **Number** | **Percentage** | **Number** |
| (1)Increase of the manual assessment | 0.0115% | 1,270 | 0.0075% | 7,190 |
| (2)Reduction of the computer processing | 18.5% | 2,042,000 | 11.7% | 11,216,800 |
| Ratio of (1) to (2) | 1:1600 | | 1:1560 | |

Note: Percentages are 0.05%∗23% for the 100GB data set and 0.05%∗15% for the 1.36TB data set.

Even though the per-page cost of the manual assessment is much greater than that of the computer processing, the cascaded structure of the recall-assured classifiers is efficient enough in terms of total processing cost. Therefore it is worthwhile considering to use it in the investigated system. The actual cost, however, should be considered when making the final decision.

## 6.3 Conclusion

To reduce the processing time, it is very important to reduce the amount of candidate pages by using the rough filtering as the first processing step because it will reduce the processing cost of the subsequent accurate classification.

The amount of pages that need computer processing for the accurate classification is reduced not only by using two types of classifiers—recall-assured and precision-assured classifiers—but also by the cascaded structure of the recall-assured classifiers. For both the 100GB and the 1.36TB data sets, the cascaded structure reduces the amount of computer processing pages by about 40%.

Even though the use of the cascaded structure increases the manual assessment a little, the increase is very small compared with the reduction of computer processing cost. The total processing cost is evidently decreased by using the cascaded structure of recall-assured classifiers.

# Chapter 7

# Conclusion and Perspective

## 7.1 Overall Conclusions

In this research, we have exploited a method for building a high-quality homepage collection by considering page group structures and have used researchers' homepages as an example of homepage categories that can be collected. After the related work was reviewed, the objectives and overall goal of the investigated system were presented in a discussion of the requirements for the target collection. Then the configuration of the overall system was presented and the methods on the rough filtering and the accurate classification were discussed in detail.

In the chapter on the rough filtering we described a method for comprehensively gathering all the probable researchers' homepages from the web while gathering as few noise pages as possible. We proposed a method using property-based keyword lists combined with four page group models (PGMs). Two original key techniques were used to reduce the number of irrelevant keywords to be propagated by exploiting the mutual relations between the content and the structures of the pages in a logical page group. The method was evaluated by comparing it with a single-page-based method in experiments using the 100GB and 1.36TB data sets as well as a manually created sample data set with various parameters. The output from the rough filtering was 23% of the 100GB data set and 15% of the 1.36TB data set. The experiment results show that it could successfully reduce the increase of the amount of gathered pages to an allowable level despite its use of a page-group-based method, which

generally causes much noise. The proposed method was also shown to be able to gather a significant number of positive pages that could not be gathered with a single-page-based method. Therefore the method is considered to have fulfilled the goal set for the rough filtering.

In the chapter on the accurate classification, we described a method for classifying the candidate pages output from the rough filtering to three classes—assured positive, assured negative, and uncertain—with high recall as well as high precision. The high performance is achieved with Support Vector Machine (SVM) by utilizing the features obtained from the textual contents, including plain-text-based and tagged-text-based features, of each page and its surrounding pages and considering their connection types ( in-linked pages, out-linked pages, and directory entry pages) in combination with the relative location of the pages (in the same, lower, and upper directories of the URL level). The feature subsets on surrounding pages are concatenated independently to improve the classification performance. Compared with the effectiveness of the baseline, the effectiveness of the proposed features in achieving higher performance was shown by experiment results. To compare our method with previous methods, we applied the proposed features and feature sets to the Web->KB data set and the obtained experiment results showing our method is effective not only for gathering researchers' homepages written in Japanese but also for gathering other categories of homepages in other languages. The analysis on classification result examples further shows that the classifier is effective in classifying difficult pages.

To reduce the manual assessment cost, we use a recall-assured classifier and a precision-assured classifier in combination and to tune them independently. The amount of pages that need manual assessment with our proposed method can be evidently reduced in comparison to the baseline (e.g. the reduction is 57.3% at 99% precision and 95% recall). We also proposed to use a cascaded structure of the recall-assured classifiers that, for both 100GB and 1.36TB data sets, reduces the computer processing cost from that for using a single recall-assured classifier by about 40%. Therefore the method is considered to have also fulfilled the goal set for the accurate classification.

One of our unique contributions is that we pointed out the importance of assuring the quality of web page collections and proposed a realistic framework to assure that quality in a process with two steps: rough filtering and accurate classification. Another is that we introduced an idea of page group models (PGMs) and demonstrated the effectiveness of using them for both filtering and classifying web pages.

- In the rough filtering, we contributed two original key techniques used in the modified PGMs to reduce irrelevant keywords to be propagated. One is to introduce a threshold for the number of out-linked page number in the same and lower directories, and the other is to introduce keyword list types and to propagate only the organization-related keyword lists from the upper directories.

- In the accurate classification, we contributed mainly in two aspects. One is to exploit features from the surrounding pages and concatenate them independently to improve the classification performance. The other is the framework for minimizing the amount of manual assessment needed to assure the required recall and precision.

Although we have not applied our method to other categories, the results of our experiments on the Web->KB data set convinced us that our method is effective in gathering many categories of research-related homepages. Although we are not sure that it is also applicable for shopping, product catalogs, and so on, we expect it to be applicable whenever only a single entity is described on each entry page. In terms of information service, the high-quality collections built with our method will be applicable to various domain-specific search engines with guaranteed high quality.

For the sample data, the analysis of classification result examples shows that there are still many misjudgment exist in the sample data set. After cleaning it, we will make it open to the researchers who need it.

For the resulting collection of researchers' homepages, since we cannot perform manual assessment at present, the collection cannot be built with guaranteed quality.

Since we fully exploit the surrounding pages, it is not suitable for providing a domain-specific search engine only by the entry pages. For instance, a researcher's name may not appear in the entry page. There is a possibility to use it in combination with a general search engine so that it can rank highly confident search results higher when a researcher's name is given as a query.

In conclusion, we investigated a two-step-processing method for building a high-quality homepage collection by considering the page group structures. The rough filtering and the accurate classification can guarantee the required quality of the homepage collection, and the processing cost can be reduced to a reasonable level.

Our method can be used not only to build domain-specific search engines with guaranteed qualities and coverage but also to compile databases and organize existing information resources.

## 7.2 Perspectives and Future Work

We have investigated a method for building a high-quality collection of homepages. Although our method is efficient and effective enough to fulfill the objectives of the research to a certain degree, it is not yet satisfactory. We will therefore continue to work on the following issues:

For the rough filtering,

- Find a more systematic way of modifying the property-based keywords and the property set.
- Try combinations of individual keyword lists with other possible keyword types in a systematic way.

For the accurate classification,

- Exploit various features on other promising clues, such as file types of link target, anchor text, and page tag structure.
- Further investigate ways to estimate the likelihood of the component pages by taking into consideration how people use the surrounding pages for judging the useful pages and introduce it to the current method.

We will also implement the investigated method in a fully operational system and further utilize the information from the homepage collection for practical applications.

In summary, tackling the diversity of web data by exploiting their rich web-based features while pursuing very high performance with little processing cost is a challenging problem. The method presented in this dissertation is considered to give a general framework for solving the same kind of problems and we hope that it will contribute to the progress of related research on web information utilization.

# Bibliography

[1] W. Koehler. Digital Libraries and World Wide Web Sites and Page Persistence. Journal of Information Research, Vol. 4(4), Jul. 1999.

[2] J. Dean and M. R. Henzinger. Finding Releated Pages in the World Wide Web. In Proc. of 8th World Wide Web Conference, Toronto, May 1999.

[3] J. R. Haritsa. The Web is the Database. In Proc. of International Workshop DNIS 2000 on Databases in Networked Information Systems, LNCS 1966, pp.91–106, Aizu, Japan, Dec. 2000.

[4] G. Arocena and A. Mendelzon. WebOQL: Restructuring Documents, Databases and Webs. In Proc. of 14th International Conference on Data Engineering, Orlando, Florida, USA, Feb. 23–27, 1998.

[5] D. Florescu, A. Levy, and A. Mendelzon. Database Techniques for the World Wide Web: A Survey. SIGMOD Record, Vol. 27(3), 1998.

[6] C. S. Baptista, F. Q. Pinto, A. Kemp, and N. Ryan. MetaCRIS: Metadata for Research Digital Libraries. In Proc. of European Conference on Current Research Information Systems (CRIS 2000), Helsinki, Finland, May 2000.

[7] A. S. Lopatenko and M. V. Kulagin. Current Research Information Systems and Digital Libraries. In Proc. of Digital Libraries: Advanced Methods and Technologies, Digital Collections. The Third All-Russian Scientific Conference. Petrozavodsk, Sep. 11–13, 2001.

[8] M. Grotschel and L. Lugger. Scientific Information Systems and Metadata, Classification in the Information Age. In Proc. of 22nd Annual Gesellschaft fur Klassifikation Confenrence (GfKl), Dresden, Mar. 4–6, 1998.

[9] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings. Keyword Extraction from the Web for FOAF Metadata. In Proc. of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web, pp.1–8, Galway, Ireland, Sep. 2004.

[10] D. B. Horn, T. A. Finholt, J. P. Birnholta, D. Motwani, and S. Jayaraman. Understanding CSCW: Looking from Above: Six Degrees of Jonathan Grudin: a Socail Network Analysis of the Evoluation and Impact of CSCW Research. In Proc. of the 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, Illinois, USA, Nov. 6–10, 2004.

[11] Y. Matsuo, H. Tomobe, K. Hasida, and M. Ishizuka. Mining Social Network of Conference Participants from the Web. In Proc. of 2003 IEEE/WIC Internatioal Conference on Web Intelligence (WI2003), Halifax, Canada, Oct. 2003.

[12] L. Terveen and D. W. McDonald. Social Matching: A Framework and Research Agenda. In Transactions on Computer-Human Interaction (TOCHI), Vol. 12(3), Sep. 2005. ACM Press.

[13] R. Bekkerman and A. McCallum. Disambiguating Web Appearances of People in a Social Network. In Proc. of 14th International Conference on World Wide Web (WWW 2005), Chiba, Japan, May 10–14, 2005.

[14] N. Doring. Personal Home Pages on the Web: A Review of Research. Journal of Computer-Mediated Communication, Vol. 7(3), 2002. (online available from http://jcmc.indiana.edu/vol7/issue3/doering.html)

[15] X. Gao, M. Zhang, and P. Andreae. Learning Information Extraction Pattern from Tabular Web Pages without Manual Labelling. In Proc. of the IEEE/WIC International Conference on Web Intelligence, Beijing, China, 2003.

[16] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, Vol. 46(5), pp.604–632, 1999.

[17] V. Dubois, M. Quafafou, and B. Habegger. Mining Crawled Data and Visualizing Discovered Knowledge. In Proc. of 1st Asian-Pacific Conference on Web Intelligent (WI2001), pp.493–497, Maebashi, Japan, Oct. 23–26, 2001.

[18] W. Lam. Intelligent Content-Based Document Delivery via Automatic Filtering Profile Generation. International Journal of Intelligent Systems, Vol. 14, pp.963–979, 1999.

[19] S. Dumais and H. Chen. Hierarchical Classification of Web Content. In Proc. of 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2000), Authens, Greece, Jul. 2000.

[20] F. Sebastiani. Machine Learning in Automated Text Categorization. Journal of ACM Computeing Surveys, Vol. 34(1), pp.1–47, March 2002.

[21] S. Lin and J. Ho. Discovering Informative Content Blocks from Web Documents. In Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Minig (SIGKDD'02), PP.588–593, Edmonton, Alberta, Canada, Jul. 23–26, 2002.

[22] O.-W. Kwon and J.-H. Lee. Web Page Classification Based on k-Nearest Neighbor Approach. In Proc. of 15th International Workshop on Information Retrieval with Asian Languages, pp.9–15, HongKong, China, Nov. 2000.

[23] Y. Wang and M. Kitsuregawa. Enhancing Contents-Link Coupled Web Clustering and Its Evaluation. DEWS2004 5-B-05, Japan, 2004. (online available from http://www.ieice.org/iss/de/DEWS/proc/2004/paper/5-B/5-B-05.pdf)

[24] P. Calado, M. Cristo, E. Moura, N. Ziviani, B. Ribeiro-Neto, and M. A. Goncalves. Combining Link-based and Content-based Methods for Web Document Classification. In Proc. of 12th International Conference on Information and Knowledge Management (CIKM'03), New Orleans, Louisiana, USA, Nov. 3–8, 2003.

[25] W. Wibowo and H. E. Williams. Simple and Accurate Feature Selection for Hierarchical Categorisation. In Proc. of Symposium on Document Engineering (DocEng'02), McLean, Virginia, USA, Nov. 2002.

[26] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pen-nock, and G. W. Flake. Using Web structure for classifying and describing Web pages. In Proc. of 11th International World Wide Web Conference, pp.562–569, Honolulu, Hawaii, USA, 2002.

[27] M. Ester, H.-P. Kriegel, and M. Schubert. Web Site Mining: A New Way to Spot Competitors, Customers and Suppliers in the World Wide Web. In Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD02), Edmonton, Alberta, Canada, Jul. 23–26, 2002.

[28] A. Arasu. Extracting Structure Data form Web Pages. In Proc. of SIGMOD2003, San Diego, CA, June 9–12, 2003.

[29] E. S. Boese and A. E. Howe. Effects of Web Document Evolution on Genre Classification. In Proc. of 14th ACM International Conference on Information and Knowledge Management (CIKM'05), Bremen, Germany, Oct. 31–Nov. 5, 2005.

[30] R. Song, H. Liu, J. Wen, and W. Ma. Learning Block Importance Models for Web Pages. In Proc. of 13th World Wide Web Conference (WWW2004), New York, NY, USA, May 17–22, 2004.

[31] C. H. Lee, M.-Y. Kan, and S. Lai. Stylistic and Lexical Co-training for Web Block Classification. In Proc. of 6th ACM International Workshop on Web Information and Data Management (WIDM'04), Washington, DC, USA, Nov. 12–13, 2004.

[32] L. K. Shih and D. R. Karger. Using URLs and Table Layout for Web classification Tasks. In Proc. of 13th World Wide Web Conference (WWW2004), pp.193–202, New York, NY, USA, 2004.

[33] W. Xi, E. A. Fox, R. P. Tan, and J. Shu. Machine Learning Approach for Homepage Finding Task. In Proc. of 9th International Symposium SPIRE, pp.145–159, Lisbon, Portugal, Sep. 11–13, 2002.

[34] M.-Y. Kan. Web Page Categorization without the Web Page. In Proc. of 13th World Wide Web Conference (WWW2004), New York, NY, USA, May 17–22, 2004.

[35] W. Kraaij, T. Westerveld, and D. Hiemstra. The Importance of Prior Probabilities for Entry Page Search. In Proc. of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02), pp.27–34, Tampere, Finland, Aug. 11–15, 2002.

[36] H.-Y. Kao, S.-H. Lin, J.-M. Ho, and M.-S. Chen. Entropy-Based Link Analysis for Mining Web Informative Structures. In Proc. of 11th ACM International Conference on Information and Knowledge Management (CIKM'02), pp.574–581, McLean, Virginia, USA, Nov. 4–9, 2002.

[37] D. Davidov, E. Gabrilovich, and S. Markovitch. Paramenterized Generation of Labeled Datasets for Text Categorization Based on a Hierarchical Diectory. In Proc. of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04), pp.250–257, Sheffield, South Yorkshire, UK, Jul. 25–29, 2004.

[38] G. Adami, P. Avesani, and D. Sona. Clustering Documents in a Web Directory. In Proc. of 5th ACM International Workshop on Web Information and Data Management (WIDM'03), pp. 66–73, New Orleans, Louisiana, USA, Nov. 7–8, 2003.

[39] V. Krikos, S. Stamou, and P. kokosis. DirectoryRank: Ordering Pages in Web Directories. In Proc. of 7th Annual International Workshop on Web Information and Data Management (WIDM'05), pp. 17–22, Bremen, Germany, Nov. 5, 2005.

[40] S. Slattey and M. Craven. Combining Statistical and Relational Methods for Learning in Hypertext Domains. In Proc. of 8th International Conference on Inductive Logic Programming, Madison, Wisconsin, USA, Jul. 1998.

[41] J. Furnkranz. Exploiting Structural Information for Text Classification on the WWW. Journal of Intelligent Data Analysis, pp.487–498, 1999.

[42] R. Ghani, S. Slattery, and Y. Yang. Hypertext Categorization Using Hyperlink Patterns and Meta Data. In Proc. of 18th International Conference on Machine Learning (ICML-01), pp.178–185, WIlliams College, US, 2001.

[43] H.-J. Oh, S. H. Maeng, and M.-H. Lee. A practical Hypertext Categorization Methos Using Links and Incrementally Available Class Information. In Proc. of 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM press, pp.264–271, Athens, Greece, Jul., 2000.

[44] S. Chakrabarti, M. H. Berg, and B. E. Dom. Distributed Hypertext Resource Discovery Through Examples. In Proc. of 25th International Conference on Very Large Data Bases, pp.375–386, Edinburgh, Scotland, UK, Sep. 1999.

[45] M. Kitsuregawa, I. Pramudiono, Y. Ohura, and M. Toyoda. Some Experimences on Large Scale Web Mining. In Proc. of 2nd International Workshop on Databases in Networked Information Systems, pp.173–178, London, UK, 2002.

[46] M. Craven, D. DiPasquo, D. Freitag, and A. McCallum. Learning to Extract Symbolic Knowledge from the World Wide Web. In Proc. of 15th Conference of the American Association for Artificial Intelligence (AAAI-98), Madison, Wisconsin, Jul. 26–30, 1998.

[47] Y. Yang, S. Slattery, and R. Ghani. A Study of Approaches to Hypertext Categorization. In Journal of Intelligent Information Systems, Kluwer Academic Press, vol. 18, pp.219–241, 2002.

[48] A. Sun and E.-P. Lim. Web Unit Mining: Finding and Classifying Subgraphs of Web Pages. In Proc. of International Conference on Information and Knowledge Management (CIKM2003), pp.108–115, New Orleans, Louisiana, USA, 2003.

[49] K. Tajima, K. Hatano, T. Matsukura, and R. Sano. Discovery and Retrieval of Logical Information Units in Web. In Proc. of Workshop on Organizing Web Space (WOWS'99), in conjunction with ACMDL'99, Berkeley, CA, USA, Aug. 1999.

[50] K. Matsuda and T. Fukushima. Task-oriented World Wide Web Retrieval by Document Type Classification. In Proc. of 8th International Conference on Information and Knowledge Management, pp.109–113, Missouri, United States, 1999.

[51] J. Cho and S. Roy. Impact of Search Engines on Page Popularity. In Proc. of WWW2004, pp.20–29, New York, NY, USA, May 17–22, 2004.

[52] A. Ntoulas, J. Cho, and C. Olston. What's New on the Web? The Evolution of the Web from a Search Engine Perspective. In Proc. of 13th World Wide Web Conference (WWW2004), pp.1–12, New York, NY, USA, May 17–22, 2004.

[53] A. Aizawa and K. Oyama. A Fast Linkage Detection Scheme for Multi-Source Information Integration. In Proc. of International Workshop on Challenges in Web Information Retrieval and Integration (WIRI 2005), pp.30–39, Tokyo, Japan, 2005.

[54] S. Chakrabarti. Mining the Web: Discovery Knowledge from Hypertext Data. Morgan Kaufmann Series in Data Management Systems, Mongan Kaufmann Publishers, 2002.

[55] R. Kosala and H. Blocheel. Web Mining Research: A Survey. SIGKDD Explorations, Vol. 2(1) pp.1–15, 2000.

[56] R. Cooley, B. Mobasher, and J. Srivasava. Web Mining: Information and Pattern Discovery on the World Wide Web. In Proc. of IEEE International Conference Tools with AI, pp.558–567, Newport, Beach, CA, 1997.

[57] B. Liu and K. C-C. Chang. Editorial: Special Issue on Web Content Mining. ACM SIGKDD Explorations, Vol. 6(2), pp.1–4, 2004.

[58] M. Harada, S. Sato, and K. Kazama. Finding Authoritative People from the Web. In Proc. of 4th ACM/IEEE-CS Joint Conference on Digital Libraries (JKDL'04), Tucson, Arizona, Jun. 7–11, 2004.

[59] J. Shakes, M. Langheinrich, and O. Etzioni. Dynamic Reference Sifting: A Case Study in the Homepage Domain. In Proc. of 6th International World Wide Web Conference, pp.189–200, Santa Clara, CA, Apr. 7–11, 1997.

[60] J. Artiles, J. Gonzalo, and F. Verdejo. A Testbed for People Searching Strategies in the WWW. In Proc. of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, Aug. 15–19, 2005.

[61] I.-H. Kang and G. Kim. Query Type Classification for Web Document Retrieval. In Proc. of 26th Annual International ACM SIGIR Conference (SIGIR'03), Toronto, Canada, Jul. 28–Aug. 1, 2003.

[62] S. Chakrabarti, M. Berg, and B. Dom. Focused Crawling: A New Approach to Topic-specific Web Research Discovery. In Proc. of 8th International World Wide Web Conference (WWW8), Toronto, Canada, May 11–14, 1999.

[63] S. Oyama, T. Kokubo, and T. Ishida. Domain-Specific Web Search with Keyword Spices. IEEE Transactions on Knowledge and Data Engineering, pp.17–27, Jan. 2004.

[64] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings. Keyword Extraction from the Web for Personal Metadata Annotation. ISWC Workshop Notes VIII

(W8)–4th International Workshop on Knowledge Markup and Semantic Annotation (Semannot2004) (in conjunction with 3rd Int'l Semantic Web Conference (ISWC2004)), pp.51–60, Hiroshima, Japan, Nov. 2004.

[65] Y. Zhang, N. Z.-Heywood, and E. Milios. Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora. In Proc. of 7th ACM International Workshop on Web Information and Data Management (WIDM'05), Bremen, Germany, Nov. 2005.

[66] K. Khan and C. Locatis. Searching through Cyberspace: The Effects of Link Display and Link Density on Information Retrieval from Hypertext on the World Wide Web. Journal of The American Society for Information Science, Vol. 49(2), pp.176–182, 1998.

[67] Y. Liu, C. Wang, M. Zhang, and S. Ma. Web Data Cleansing for Information Retrieval Using Key Resource Page Selection. In Proc. of 14th International World Wide Web Conference (WWW2005), Chiba, Japan, May 10–14, 2005.

[68] W. Li, K. Candan, Q. Vu, and D. Agrawal. Retrieving and Organizing Web Pages by "Information Unit". In Proc. of 10th International World Wide Web Conference (WWW10), Hongkong, May 1–5, 2001.

[69] T. Masada, A. Takasu, and J. Adachi. Improving Web Search Performance with Hyperlink Information. IPSJ Transactions on Databases, Vol. 46(8), pp.48–59, 2005.

[70] S. Chakrabarti. Data Mining for Hypertext: A Tutorial Survey. ACM SIGKDD Explorations, Vol. 1(2), pp.1–11, 2000.

[71] R. Bekkerman, R. Ei-Yaniv, N. Tishby, and Y. Winter. Distributional Word Clusters vs. Words for Text Categorization. In Journal of Machine Learning Research, Vol. 3, pp.1183–1208, 2003.

[72] A. Sun, E.-P. Lim, and W.-K. Ng. Web Classification Using Support Vector Machine. In Proc. of 4th International Workshop on Web Information and Data Management, ACM Press, pp.96–99, McLean, Virginia, USA, 2002.

[73] J. Sun, B. Zhang, Z. Chen, Y. Lu, C. Shi, and W. Ma. GE-CKO: A Method to Optimize Composite Kernels for Web Page Classification. In Proc. of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI2004), pp.299–306, Beijing, China, 2004.

[74] M.-Y. Kan and H.O.N. Thi. Fast Webpage Classification Using URL Features. In Proc. of 14th ACM International Conference on Information and Knowledge Management (CIKM'05), pp.325–326, Bremen, Germany, 2005.

[75] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. In Proc. of International Conference Management of Data (SIGMOD'98), pp.307–318, Seattle, WA, USA, 1998.

[76] M. Chau. Applying Web Analysis in Web Page Filtering. In Proc. of ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'04), pp.376, Tucson, Arizona, USA, 2004.

[77] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. Overview of the Web Retrieval Task at the Third NTCIR Workshop. NII Technical Report, No.NII-2003-002E, NII, 2003.

[78] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. Evaluation Methods for Web Retrieval Tasks Considering Hyperlink Structure. IEICE Transactions on Information and Systems, Vol. E86-D(9), pp.1804–1813, 2003.

[79] K. Oyama, E. Ishida, and N. Kando. In Proc. of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering (Sep. 2001–Oct. 2002). National Institute of Informatics, Tokyo, 2003. (online available from http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/index.html)

[80] N. Kando and H. Ishikawa. In Proc. of the Fourth NTCIR Workshop on Research in Information Access Technologies, Information Retrieval, Question Answering and Summarization (Apr. 2003–Jun. 2004), National Institute of Informatics, Tokyo, 2005. (online available from http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings4/index.html)

[81] K. Oyama, M. Takaku, H. Ishikawa, A. Aizawa, and H. Yamana. Overview of the NTCIR-5 WEB Navigational Retrieval Subtask 2. In Proc. of NTCIR-5 Workshop Meeting, Tokyo, Japan, Dec. 6–9, 2005. (online available from http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings5/data/WEB/NTCIR5-OV-WEB-OyamaK.pdf)

[82] D. Hawking and N. Craswell. Overview of the TREC-2001 Web Track. Proc. NIST Special Publication 500–250: The Tenth Text Retrieval Conference (TREC 2001). (online available from http://trec.nist.gov/pubs/trec10/papers/web2001.ps.gz)

[83] K. Oyama, K. Eguchi, H. Ishikawa, and A. Aizawa. Overview of the NTCIR-4 WEB Navigational Retrieval Task 1. In Proc. of NTCIR-4 Workshop Meeting, Tokyo, Japan, Jun. 2–4, 2005. (online available from http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings4/index.html)

[84] K. Frantzi, S. Ananiadou, and H. Mima. Automatic Recognition of Multi-word Terms: The C-value/NC-value Method. International Journal on Digital Libraries, Vol. 3(2), pp.115–130, Aug. 2000.

[85] I. Witten. Browsing around a Digital Library. In Proc. of the Australasion Computer Science Conference, pp.1–14, Auckland, Australia, 1999.

[86] Y. ZHANG, N. Zincir-Heywood, and E. Milios. Term-Based Clustering and Summarization of Web Page Collections. In Advances in Artificial Intelligence, Proc. of 7th Conference of the Canadian Society for Computational Studies of Intelligence, pp.60–74, London, ON, Canada, May 2004.

[87] E. Milios, Y. Zhang, B. He, and L. Dong. Automatic Term Extraction and Document Similarity in Special Text Corpora. In Proc. of 6th Conference of the Pacific Association for Computational Linguistics, pp.275–284, Halifax, NS, Canada, Aug. 2003.

[88] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. N.-Manning. KEA: Practical Automatic Keyphrase Extraction. In Proc. of 4th ACM Conference on Digital Libraries, pp.254–255, Berkeley, CA, USA, Aug. 1999.

[89] A. Berger and V. Mittal. OCELOT: A System for Summarizing Web Pages. In Proc. of 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.144–151, Athens, Greece, Jul. 2000.

[90] T. Joachims. A Statistical Learning Model of Text Classification for Support Vector Machines. In Proc. of SIGIR'01, New Orleans, Louisiana, USA, Sep. 9–12, 2001.

[91] Y. Yang and X. Liu. A Re-examination of Text Categorization Methods. In Proc. of SIGIR'99, 22nd ACM International Conference on Research and Development in Information Retrieval, pp.42–49, Berkeley, US, 1999.

[92] T. M. Cover and J. A. Thomas. Chapter 2, Entropy, Relative Entropy and Mutual Information. Elements of Information Theory. Wiley, 1991.

[93] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to Construct Knowledge Bases from the World Wide Web. In Journal of Artificial Intelligence, Vol. 118(1–2), Special Issue on Intelligence Internet Systems, pp. 69–113, Apr. 2000. Elsevier Science Publishers Ltd.

# List of Publications

## Publications and Transactions

### Transactions and Journals

1. Yuxin Wang and Keizo Oyama. Combining Page Group Structure and Content for Roughly Filtering Researchers' Homepages with High Recall. IPSJ Transactions on Databases, Vol. 47, No. SIG 8 (TOD 30), pp.11–23, Jun. 2006.

### Conference Proceedings

1. Yuxin Wang and Keizo Oyama. A Method for Creating a High Quality Collection of Researchers' Homepages from the Web. In Proceedings of the 8th International Conference on Asian Digital Libraries (ICADL2005), LNCS 3815, pp. 473–474, Bangkok, Thailand, Dec. 2005. (poster)
2. Yuxin Wang and Keizo Oyama. Combining Page Group Structure and Content for Roughly Filtering Researchers' Homepages with High Recall. In Proceedings of DBWeb2005, pp. 189–196, Tokyo, Japan, Nov. 2005.
3. Yuxin Wang and Keizo Oyama. A Method for Creating a High Quality Collection of Researchers' Homepages from the Web. In Proceedings of Data Engineering Workshop 2005 (DEWS2005), Sasebo, Japan, Feb.28–Mar.2, 2005. (online, available from http://www.digitalcity.gr.jp/ kani/DEWS/papers/5C-i10.pdf)
4. Yuxin Wang and Keizo Oyama. Webpage Classification Exploiting Contents of Surrounding Pages for Building a High-quality Homepage Collection. The 9th International Conference on Asian Digital Libraries (ICADL2006), Kyoto, Japan, Nov. 27th–30th, 2006. (accepted as a short paper)

## Presentations and Posters

1. Yuxin Wang and Keizo Oyama. A Method for Collecting Unspecified Researchers' Homepages Utilizing Web-Specific Features. Information and System Society of IEICE, Special session, Student Poster D-SP-30, p47, Osaka, Japan, 2005.

2. Yuxin Wang and Keizo Oyama. A Method for Collecting Unspecified Researchers' Homepages Utilizing Web-Specific Features. NII Open House, National Institute of Informatics, Tokyo, Japan, Jun. 2005.

# Appendix A

# Table Actual Keywords

| Type | Keyword lists | Keywords |
|---|---|---|
| Non-organization-related | general word | 研究 |
| | research topic | 研究内容, 研究対象, キーワード, 課題, テーマ, 研究活動, 研究紹介 |
| | title | 博士, 教授, 講師, 助手, 技官, 研究員, リサーチアソシエート, ポスドク, ポストドクター |
| | position | 資格, 職名, 役職, 現職 |
| | history | 略歴, 学歴, 職歴, 学位, 履歴, 卒業, 経歴, プロフィール, profile, 自己紹介, 出身 |
| | achievement | 論文, 報告, 著書, 共著, 出版, 発表, 賞, 業績, プロジェクト, 研究成果, フィールドワーク, 特許, フィルドワーク, paper, 翻訳, 著作, 講演, 学会誌 |
| | lecture | 講義, ゼミ, 授業, クラス, 科目, 演習, 教材 |
| | academic society | 学会, 会員, 協会, 研究会 |
| Organization-related | major | 専門, 分野, 学科, 専攻 |
| | member | メンバー, 教官, スタッフ, 職員, 研究者, 教員, 構成員 |
| | organization | 大学, 研究所, 研究センター |
| | section | 所属, 部門, 講座, 研究室, 実験室, lab, laboratory, 学群, 学系, 学部, 研究科 |