IPSJ Transactions on Databases Vol. 2 No. 2 29–43 (June 2009)

Regular Paper

Web Page Classification Based on Surrounding Page Model Representing Connection Type and Directory Hierarchy

Yuxin Wang[†1,*1] and Keizo Oyama[†2]

We propose a web page classification method that is suitable for building web page collections and show its effectiveness through experimentation. First, we describe a model that represents a surrounding page group structure that takes the link relation and directory hierarchy relation into consideration and a method for extracting features based on the model. The method is tested through classification experimentation on two data sets and using the support vector machine (SVM) as the classification algorithm, and its effectiveness is confirmed through comparison with a baseline and the results of previous studies. The contribution of each part of the surrounding pages is also analyzed. Next, we test the method's performance on overall recall-precision range and find that it is superior in the high recall range. Finally, we estimate the performance of a three-grade classifier composed with the method and the amount of manual assessment required to build a web page collection.

1. Introduction

The Web is becoming more and more important as an information source for various information services. To make use of it first requires building of a web page collection of a given category with a guaranteed level of quality (i.e., high recall and high precision). However, even with the current state-of-the-art classification techniques, this task requires a large amount of manual assessment because of the diversity in styles, granularities, and structures of web pages, the vastness of the data, and the sparseness of relevant pages. Therefore, it is crucial to develop a method for automatically selecting as many pages as possible that are confidently relevant and irrelevant, so that the number of uncertain pages can be minimized. One solution to this problem is to compose a three-grade classifier by combining two binary classifiers, each satisfying a given recall or a given precision.

Various web page classification techniques have been studied to improve basic performance under certain trade-offs between recall and precision (e.g., the harmonic mean, known as the F-1 measure, the break-even point of recall and precision, etc.). However, only a small amount of attention has been paid to classifications when a high recall or a high precision is given as a condition, namely recall/precision-controlled performance. In this regard, our goal is to develop a web page classification method that is not only effective in terms of basic performance but also in terms of quality-controlled performance.

The main difficulty in web page classification comes from the diversity of granularities and structures of web pages. It is not rare that the target information is fragmented into separate pages and their entry page provides only hyperlinks to those pages. Furthermore, important pieces of information may be presented only on the entry page's parent page. To take a search for a researcher's homepage as an example, a clue is whether there is information about a university or a research institute on a page in the upper directory and whether the bibliography is on an out-link page. Such information is crucial especially when the researcher's entry page provides, e.g., his/her name, photo, and hyperlinks (with image anchors) to component pages, but no other information.

To achieve our goal, especially for recall-controlled performance, we have to be able to efficiently deal with such entry pages. We approach this problem by exploiting information in the surrounding pages as well as in the entry page itself.

Ideally, we should use the surrounding pages according to their semantic relations with the entry page, but estimating such relations is a hard task. Thus, we attempted an alternative approach, which is to use formal relations, such as link relations and directory structures, to capture the semantic relation.

Previous studies that have tried a similar approach got only marginal gains. This was probably because they only considered the link relations and overlooked the directory structures, or at most dealt with them separately. We combine them

†1 The University of Tokyo
†2 National Institute of Informatics
∗1 Presently with Team Lab Corp.
⋆1 The work presented in this paper was partly done when Ms. Wang was with the National Institute of Informatics.

in a way that allows the semantic relations to be captured in some regard.

Approximately speaking, we use two types of relations between an entry page and a surrounding page, i.e., the connection type and the directory hierarchy level, corresponding to the above-mentioned link relations and directory structures, respectively. An entry page and its surrounding pages are divided into several surrounding page groups according to certain conditions set with these relations. Then, we extract features from each surrounding page group and compose a feature vector consisting of them all.

We used two types of textual feature in our study, but various other features and/or feature metrics from other studies can be used as well.

We evaluated the features through experimentation with Web-KB ⋆1, an English data set, and ResJ-2, a Japanese data set prepared by the authors. A support vector machine (SVM) with a linear kernel was used as the classification algorithm. Note that the other commonly used classifiers mentioned in Section 2 can be also used with the proposed features.

The rest of this paper is organized as follows. Related work is introduced in Section 2. The details of the proposed method are presented in Section 3, including the concept of the surrounding page model and the feature extraction process. Section 4 describes the experiments using the two data sets and presents experimental results including a comparison with previous work. Section 5 discusses the effectiveness of the proposed feature set and its applicability to a three-grade classifier. Finally, we conclude our work in Section 6.

2. Related Work

The method proposed in this paper belongs to the web page classification domain, and it is closely related to the web page search and clustering domains. Two major problems in these domains are determining which information sources to use and how to use them.

The previous studies have tried to exploit the textual content on each page and various web-related information sources [1], such as html tags [2)–5)], URLs [3),5)–7)], subgraphs of web pages [8),9)], directory structures [9),10)], anchor texts [2)–4)], the content of globally linked pages [4),11)–13)], and the content of surrounding pages [3),8)–10),14)].

All of these information sources except the last one are used to capture features that are characteristic to the target pages, and are effective at emphasizing the highly probable pages. The last one, i.e., content of surrounding pages, is used to collect relevant information dispersed over the component pages, and it is effective for comprehensively gathering potential pages. However, this last source tends to increase the amount of noise, and no clear performance improvement has been obtained from it in the previous studies.

Nevertheless, since comprehensiveness (or recall) is the key to assuring the quality of a web page collection, we are mainly looking to exploit the surrounding pages as information sources.

The previous studies that have tried to exploit information in the surrounding pages are as follows.

Sun, et al. [9)] proposed a method to first classify each page based on its content and then to iteratively classify the web page subtrees by combining the previous results while taking into consideration other information sources, such as the link and directory structures. Although they achieved good results, their methods required extra training data for the support pages as well as the main pages. In addition, their approach does not work when an entry page contains no textual information, i.e., only hyperlinks.

Masada, et al. [8)] proposed a method to cluster web pages based on their link structures, etc., and merge the score (or weight) of each content word to generate the document vector. However, the effectiveness of this proposal was limited, probably because it also merges many irrelevant words from the linked pages.

Yang, et al. [10)] defined five hypertext regularities and tested how their presence influenced the classification performance. By exploiting the co-referencing regularity among the other regularities, they used Naïve Bayes and the k-nearest neighbor method to treat all the content words of the linked pages together and a first order inductive learner (FOIL) to treat the content words separately for every linked page. Unfortunately, the classification performance suffered for all cases.

Craven, et al. [3)] compared the FOIL with the predicate invention for large

⋆1 http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

feature spaces (FOIL-PILFS) to a FOIL using much richer features to prove the algorithm's effectiveness. However, they made no comparison between different features, and hence, the effectiveness of FOIL-PILFS in exploiting the content of linked pages could not be judged.

In Ref. 14), we defined the page group models and proposed selectively merging words from the surrounding pages according to the page group models to generate a document vector, which can then be used for keyword-based web page filtering. The basic idea of a page group model is the same as the surrounding page group of the current work, except that a page group model merges the words in all the surrounding pages with those on the entry page. The method was shown to be effective, but it still suffers from noise.

Many other studies have tested these features using content from surrounding pages, but none showed a clear performance improvement; some even degraded performance.

We used a different approach to the ones above in order to exploit the content in the surrounding pages while considering the link relations and directory structures. In addition, almost none of the previous studies tried to provide a framework for assuring the quality level required for practical applications. In the present study, we approached this problem by building a three-grade classifier composed of a recall-controlled classifier and a precision-controlled classifier, each independently tuned. We experimentally analyzed the behaviors of the classifiers at very high recall and very high precision ranges with the proposed features.

3. Surrounding Page Model and Feature Extraction

First, we shall describe the concept of the surrounding page model and the method for extracting features. Then, we shall present the feature metrics that we tested.

3.1 Surrounding Page Model

The following are explanations and definitions of terms.

An "entry page" (EP) is a physical page under consideration that consists of a single physical document (or file). We use the term "target page" instead of this word when we refer to a virtual page that represents the EP and its related pages.

A "surrounding page" (SP) is a page placed near an EP in the link and directory structures within a web site. More accurately, an SP is a page that has certain "connection type relations" and "directory hierarchy relations" with an EP.

The following is the definition for the three "connection type relations" and three "directory hierarchy relations" between an SP and an EP. Let p be an SP:

Connection type relation $R_c = \{\text{in\_link}, \text{out\_link}, \text{dir\_entry}\}$:

- $\text{in\_link}(p)$: p has a hyperlink to EP,
- $\text{out\_link}(p)$: p has a hyperlink from EP,
- $\text{dir\_entry}(p)$: p is a directory entry page;

Directory hierarchy relation $R_h = \{\text{same\_dir}, \text{upper\_dir}, \text{lower\_dir}\}$:

- $\text{same\_dir}(p)$: p resides in same directory as EP,
- $\text{lower\_dir}(p)$: p resides in lower part of directory subtree of EP,
- $\text{upper\_dir}(p)$: p resides in upper part of directory path of EP.

Then, a "surrounding page group" (SPG) is a set of SPs satisfying certain given conditions on the connection type and the directory hierarchy level.

An "element surrounding page group" (element SPG) is a set of SPs that share the same connection type relation and the same directory hierarchy relation, defined as follows:

$$S_{c,h} = \{p | c(p) \wedge h(p)\}$$

for $c \in R_c, h \in R_h$. In addition, we treat $S_{\text{entry\_page}}$, a set that consists of only an EP, as a kind of element SPG for convenience. Consequently, there are ten $(= 3 \times 3 + 1)$ element SPGs.

Let $\mathcal{S}$ be a set consisting of all element SPGs and $\mathcal{T}$ be a subset of $\mathcal{S}$. Then, SPG is formally defined as:

$$S = \bigcup_{T \in \mathcal{T}} T$$

In practice, however, we use several meaningful SPGs as follows:

$$S_c = \bigcup_{j \in R_h} S_{c,j}$$

$$S_h = \bigcup_{i \in R_c} S_{i,h}$$

$$S_{\text{all\_sp}} = \bigcup_{i \in R_c} \bigcup_{j \in R_h} S_{i,j}$$

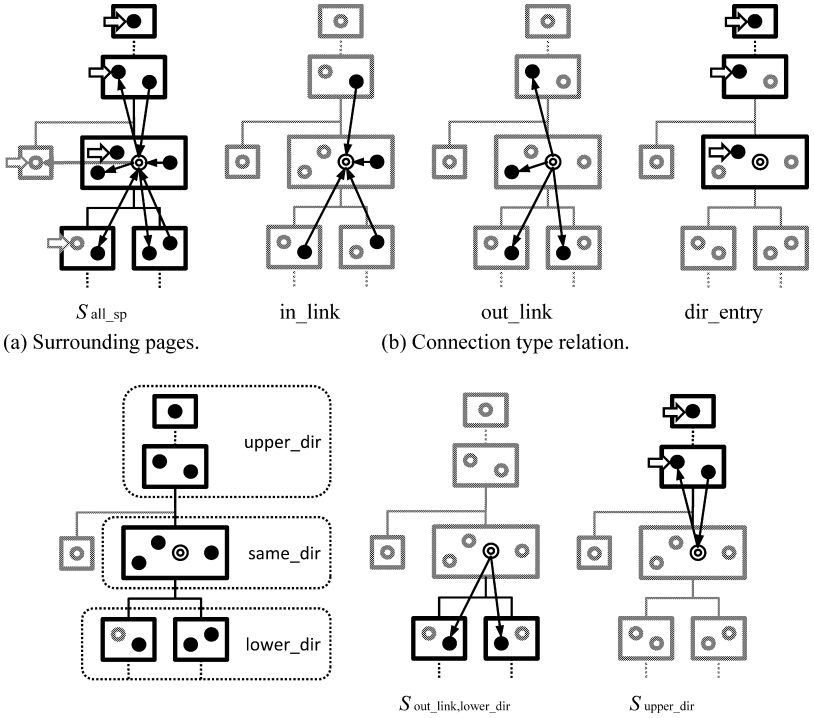


Fig. 1 Concept of surrounding page group. Double circle denotes EP, black circle denotes SP, rectangle denotes directory, thin arrow denotes hyperlink, and outline arrow denotes directory entry page.

$$S_{\text{all\_ep\_sp}} = S_{\text{entry\_page}} \cup S_{\text{all\_sp}}$$

for $c \in R_c, h \in R_h$.

Figure 1 illustrates the above-mentioned relations and presents example SPGs. Each figure presents the same pages. Figure 1 (a) shows a simple example of an EP and its SPs, which are indicated by a double circle and by black circles, respectively. Rectangles, thin arrows, and outline arrows denote directories, hyperlinks, and directory entry pages, respectively. Items in gray indicate that they are not used for the SPG, as mentioned below.

Figure 1 (b) presents the connection type relations. We use dir_entry in addition to in_link and out_link. An SP has a dir_entry relation if it is a directory entry page, which is defined as a page that is usually referred by the URL of the directory itself (i.e., URL ending with a '/' character). We added this relation because even when there is no explicit link, there is some kind of implicit relation between a directory entry page and the other pages in the same directory, and sometimes a dynamic link generated by the scripts and others exists. We judge the directory entry pages by checking if the prefix of the file name is either an "index" or "default" and if the suffix indicates a textual content and is not "rss".

Figure 1 (c) presents the directory hierarchy relations. When created by a person or an organization, the directory hierarchy usually reflects the view of the creator on the content. Therefore, we expect that hierarchical relations of, e.g., concept, organization, and composition, can be captured by dealing with SPs according to their relative hierarchy in the directory structure. We do not care about the difference in levels in the upper (or lower) directories, because it seems to make no semantic difference. We do not use sibling or collateral directories either, because the pages contained in such directories have various relations with an EP and are often irrelevant, and because we consider that they convey only a small amount of information that is useful for classification but is not included in the SPs.

Figure 1 (d) shows example SPGs. The one on the left is the element SPG $S_{\text{out\_link, lower\_dir}}$, and the one on the right is the SPG $S_{\text{upper\_dir}}$, which consists of three element SPGs: $S_{\text{in\_link, upper\_dir}}$, $S_{\text{out\_link, upper\_dir}}$, and $S_{\text{dir\_entry, upper\_dir}}$.

In view of the semantic relation with an EP, each element SPG has its own implicit meaning. For example, $S_{\text{out\_link,lower\_dir}}$ may contain detailed information on the object presented on the target page; $S_{\text{dir\_entry,upper\_dir}}$ may contain information on the organization the object belongs to. However, in the lower directory subtree, because most of meaningful directory entry pages are also out-link pages at the same time, and because out-link pages are much clearer in their meaning than directory entry pages, we decided not to use the directory entry pages. Therefore, we deem $S_{\text{dir\_entry,lower\_dir}} = \emptyset$ in the above-mentioned definitions.

Note that element SPGs are not necessarily exclusive for the following reasons. First, more than one connection type relation can hold for an SP; e.g., an SP can be an in-link page, an out-link page, and a directory entry page at the same

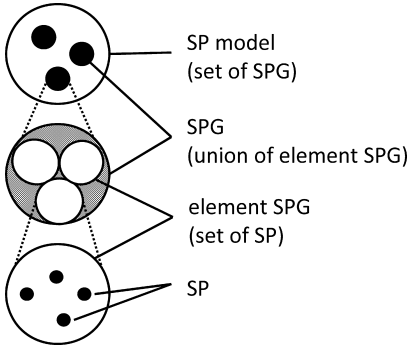


Fig. 2 Composition of surrounding page model.

time. Second, an EP can be a directory entry page. However, since such an SP is considered to convey rather important information, we allow such duplications to exist.

Now, let us introduce a "surrounding page model" (SP model), which is intended to represent the way to partition SPs into SPGs. It is formally defined as an arbitrary set of SPGs. However, for practical reasons, we add a condition that no element SPG is included in more than one of its member SPGs; i.e., let $\mathcal{M}$ be an SP model; then,

$$S_1 \cap S_2 = \emptyset, \qquad S_1, S_2 \in \mathcal{M}, \quad S_1 \neq S_2.$$

Figure 2 illustrates how SPs, element SPGs, SPGs, and SP models are related.

Although various SP models are possible and we actually tested a considerable number of them, we will discuss only the following typical SP models in this paper.

Single page model : $\mathcal{M}_{\text{single}} = \{S_{\text{entry\_page}}\}$ is used as a baseline without surrounding pages.

Monolithic page model : $\mathcal{M}_{\text{monolithic}} = \{S_{\text{all\_ep\_sp}}\}$ is used for comparison by testing a naive method used in many of the previous studies.

Consolidated page model : $\mathcal{M}_{\text{consolidated}} = \{S_{\text{entry\_page}}, S_{\text{all\_sp}}\}$ is a method similar to some of the previous studies mentioned in Section 2.

Full-structured page model : $\mathcal{M}_{\text{full}} = \mathcal{S}$ is the proposed method that uses the entire element SPGs.

3.2 Feature Extraction

We use two types of textual features: *plain* and *tagged*. The *plain* features are extracted from the full text, excluding tags, scripts, comments, etc., namely the plain text. They are tokenized with *Chasen* [⋆1] for ResJ-2 and with *Rainbow* [⋆2] for Web-KB. Then, using sample data labeled in advance, the top 2,000 terms in terms of their mutual information [15] are selected as the feature terms. Mutual information is defined by

$$I = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p_1(x) p_2(y)},$$

where $p(x, y)$ is the joint probability distribution function of X and Y, and $p_1(x)$ and $p_2(y)$ are the marginal probability distribution functions of X and Y, respectively. Here, X denotes the presence of the feature term, Y denotes the label (positive or negative), and the probability distribution function represents the fraction of the samples with the given variable(s).

The *tagged* feature is extracted from the text segments "text" that match a string pattern ">text<" or "<img···alt= "text"···>" in the full text and that are not more than 16 bytes long, omitting the spaces for ResJ-2, and not more than four words long for Web-KB, namely the tagged text segments. We do not care about the tag name in the former pattern. These are tokenized in the same way as mentioned above, and we select the terms with no less than a 1% file frequency for ResJ-2 and all the terms for Web-KB. Since the tagged text segments often contain attribute names (e.g., "<h2>Curriculum vitae</h2>" and "<table>···<th>CPU type<td>···</table>"), we believe the *tagged* features effectively represent the target page's category. Note that all the tagged text segments are also included in the plain texts, and therefore, a single word may be extracted as both *plain* and *tagged* features.

We use a feature metric that represents the existence of the feature term on any member page of a SPG (1: present, 0: absent). This is the simplest and most frequently used feature representation for document classification. Certain more

⋆1 http://chasen.naist.jp/hiki/ChaSen/
⋆2 http://www.cs.cmu.edu/~mccallum/bow/rainbow/

complicated feature metrics based on term frequency and/or document frequency are also used. However, since the purpose of this work is to investigate the use of the surrounding pages, we decided not to use them.

In the experiment described below, we report the results for the following feature vectors:

basic feature vector : consisting of *plain* features,

extended feature vector : consisting of *plain* and *tagged* features.

According to the SP model, we extract the features from the pages in each member SPG and compose a feature vector consisting of all the features. The feature vector of a target page is represented as

$$\mathbf{f} = (\mathbf{f}_1 \dots \mathbf{f}_M), \quad \mathbf{f}_i = (f_1 \dots f_{N_P} \quad f_{N_P+1} \dots f_{N_P+N_T}),$$

where M is the number of SPGs in the SP model, and N_P and N_T are the numbers of *plain* and *tagged* feature terms, respectively. Note for the *basic* feature vector, we regard $N_T = 0$.

We used an SVM with the feature vector, but other commonly used classifiers can be used instead. Moreover, by adding or replacing elements of the feature vector, many of the features and/or feature metrics proposed in other studies can be used with the SP model.

4. Experiments

4.1 Data Sets

We used Web-KB and ResJ-2 as the experimental data sets. Both were used to test the proposed features in terms of the F-measure for evaluating the basic performance. ResJ-2 was used to test the features in a wider recall-precision range and at very high recall/precision requirement for evaluating the quality-controlled performance.

4.1.1 Web-KB

Web-KB is an English data set provided by the World Wide Knowledge Base (Web-KB) Project of the CMU text learning group, and it is commonly used as a test collection for web page classification tasks. The web pages were collected from the computer science departments of 182 universities. The Project has manually classified the data into seven categories, and following the practice of the previous studies, we chose to experiment with four out of the seven categories (i.e., *course*, *faculty*, *project*, and *student*). The pages of the given category were used as positive samples, and the pages of all the other categories were used as negative samples.

As recommended by the Web-KB Project, we used the *leave one university out* cross validation method for the data from four designated universities. The pages collected from the other universities were always used as training data.

Web-KB, which consists of 8,282 web pages, is a rather small data set. Moreover, in terms of the test data for each fold of each category, the data was crawled from only two web servers on average, and more than 97% of the data came from the most dominant web servers of the respective universities. In addition, for *course*, *faculty*, and *student*, a majority of the pages shared the same directory path excluding the lowest directory element. Therefore, the context in the directory structure is rather uniform, and many SPs in the upper directories must be shared by the majority of the target pages. This situation may lead to unpredictable side effects, and it will be case-by-case whether it is advantageous or disadvantageous to our method. Therefore, we have to be careful in analyzing the experimental results.

4.1.2 ResJ-2

ResJ-2 is a Japanese data set that was manually prepared by the authors from NW100G-01 [⋆1], which is a 100-GB web page collection containing 11,038,720 pages crawled from the '.jp' domain for NTCIR-3/4 WEB Tasks [⋆2,16),17)]. We sampled the web pages containing Japanese family names, judged each page as to whether it was a researcher's homepage by looking at the entry page and, if necessary, the surrounding pages, and obtained 426 positive samples. We also judged 1% of the rough filtering output as described in Ref. 14), and obtained 534 positive samples and 20,366 negative samples. Consequently, ResJ-2 is a researcher's homepage collection containing 960 positive samples and 20,366 negative samples. Five-fold cross validation was used for the experiments.

4.2 Basic Performance Experiments

The SVM has been shown to be effective for text classification, and we used

⋆1 Available for research purposes from the National Institute of Informatics. See http://research.nii.ac.jp/ntcir/permission

⋆2 http://research.nii.ac.jp/ntcweb/

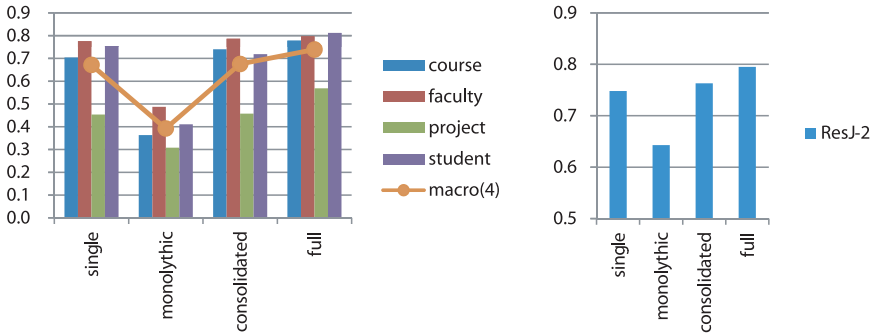


Fig. 3 Basic classification performance of proposed approach compared with that of baseline and primitive method using the *basic* feature vector.

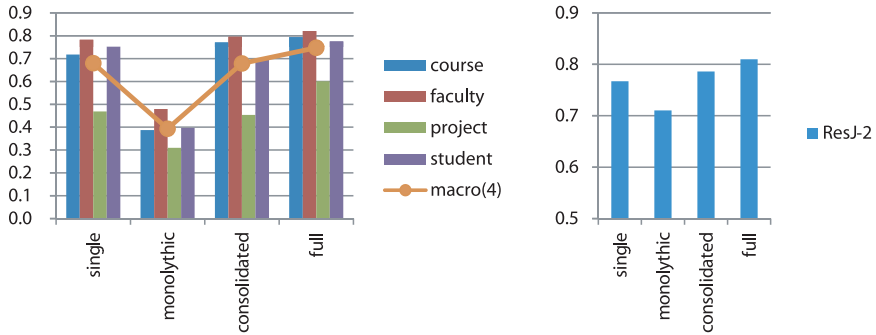


Fig. 4 Basic classification performance of proposed approach compared with that of baseline and primitive method using the *extended* feature vector.

the SVM^{light} package[⋆1] with a linear kernel. We tuned it with its options c (trade-off between training error and margin) and j (cost factor by which the training errors on positive examples out-weigh the errors on negative examples). The classification performance was evaluated by using the F-measure defined as

$$\text{precision} = \frac{|P_P \cap P_T|}{|P_P|}, \quad \text{recall} = \frac{|P_P \cap P_T|}{|P_T|},$$

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

where P_P is a set of positive predictions and P_T is a set of true positive samples.

We used a macro-averaged F-measure (i.e., average of F-measures over the four categories; referred to as "Macro(4)" below) to evaluate the performance on the Web-KB data set.

The features were extracted as described in Section 3.2. There were approximately 600 *tagged* feature terms for Web-KB and 1,200 for ResJ-2. We conducted the following classification experiments.

Experiment 1: Overall effectiveness

We compared the basic classification performances of $\mathcal{M}_{\text{full}}$ and $\mathcal{M}_{\text{consolidated}}$ with those of $\mathcal{M}_{\text{single}}$ and $\mathcal{M}_{\text{monolithic}}$ using the *basic* and *extended* feature vectors. **Figures 3** and **4** present the respective results (only the suffix of the SP model name is indicated in the figures, hereinafter).

It is clear that $\mathcal{M}_{\text{monolithic}}$ degrades the performance on all of the data sets and categories, in accordance with many previous informal reports not cited in this paper. $\mathcal{M}_{\text{consolidated}}$ performs just as well as $\mathcal{M}_{\text{single}}$; in other words, there is only a small contribution from the surrounding pages, which is in accordance with the experimental results of previous studies [3),10)]. In contrast, $\mathcal{M}_{\text{full}}$ outperforms $\mathcal{M}_{\text{single}}$ and $\mathcal{M}_{\text{consolidated}}$, demonstrating the effectiveness of our approach.

Comparing the graphs in Fig. 4 with Fig. 3, we can see that the *extended* feature vectors perform slightly better on average and in many of the individual cases, while the tendencies among the feature sets and data sets are quite similar.

Taking into consideration other experimental results not shown here, we found that the tagged text segments tend to contain typical words in each category and, although the method is quite simple, the *tagged* features contribute to the web page classification performance. Therefore, we used the *extended* feature vector in the following experiments, unless mentioned otherwise.

As mentioned in Section 4.1, since Web-KB consists of a small number of web sites, there may be some unpredictable effects. Thus, to see how the proposed method works, **Fig. 5** compares the proposed method with the baseline for each fold. $\mathcal{M}_{\text{full}}$ outperformed $\mathcal{M}_{\text{single}}$ for 14 out of 16 folds. This shows that the proposed method is rather stable.

Table 1 compares our method with some of the previous methods that had

⋆1 http://svmlight.joachims.org/

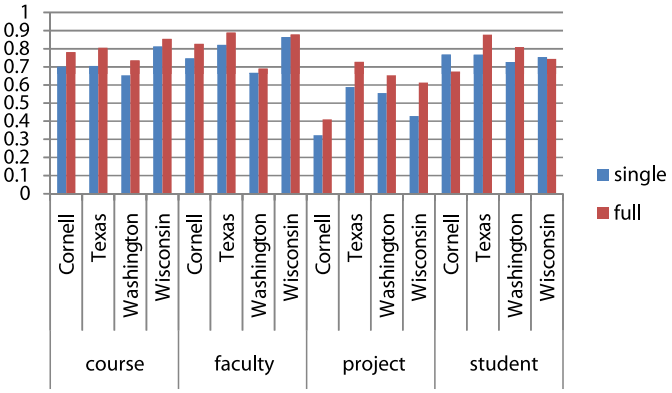


Fig. 5 Per university performance comparison.

Table 1 Classification performances of the *extended* feature vector and previous studies on Web-KB data set.

Method			*course*	*faculty*	*project*	*student*	Macro(4)
Experimented methods	compared methods	$\mathcal{M}_{\text{single}}$	.704	.776	.454	.754	.672
		$\mathcal{M}_{\text{consolidated}}$	.740	.787	.457	.719	.676
	proposed methods	$\mathcal{M}_{\text{full}}$	**.778**	**.798**	**.569**	**.812**	**.739**
Previous works	using contents of surrounding pages	SVM-iWUM (α=1)	.547	**.876**	.171	**.958**	**.638**
		FOIL (Tagged Words)					.529
		kNN (Tagged Words)					.591
	using other information sources	SVM (TA)	**.682**	.659	.325	.730	.599
		SVM-FST (XATU)	.609	.409	**.665**	.253	.484
		ME (TU)					.627
		FOIL (Linked Names)					.629

been tested with Web-KB. The best results among the experimented methods and among the previous methods are boldfaced. The previous methods that outperformed $\mathcal{M}_{\text{full}}$ are underlined.

The first three previous studies listed in Table 1 used the content of surrounding pages taking into consideration the directory and link structures, but in different ways from our approach.

"SVM-iWUM (α=1)"[9] first classifies each page based on its content using an SVM and then iterates to merge web page subtrees and to classify the results until they become stable. The merging process uses the directory structure and the previous classification results, and the classifying process uses the link and directory structures together with the previous classification results to generate statistical features. Thus, the link and directory structures are not directly combined with the content from the surrounding pages. The object of this method is not simply to classify web pages but also to find web units consisting of a key page and support pages, and accordingly it uses additional training data for the support pages. Therefore, its advantageous conditions should be taken into account in any comparison with other methods.

"FOIL (Tagged Words)"[10] used a FOIL with a relationship representing the occurrence of the terms in the entry page and in its linked pages. This means all the surrounding pages are dealt with separately. "kNN (Tagged Words)"[10] used the k-nearest neighbor method with the features from all the linked pages together as well as the features from the entry page. This means all the surrounding pages are dealt with together.

The other four used various information sources. "SVM (TA)"[2] used an SVM with the features from the page, title, and anchor texts of the entry page. "SVM-FST (XATU)"[7] used an SVM with the features from the page, title, anchor, and URL text of the entry page. "ME (TU)"[6] used a maximum entropy method with the features from the page text and the URL of the entry page. "FOIL (Linked Names)"[10] used a FOIL with a relationship representing each web page with the identifiers of the out-link pages but not their content.

The results show that our method outperformed all seven previous methods based on Macro(4). Our method outperformed nine out of 12 on a per-category basis (F-measures of the individual categories are available for only three of the previous studies).

Despite these good results, our method is not strictly comparable with the previous methods, because although the data set is the same, the method might have been evaluated on different parts of the data set or with different cross validation methods. However, although our method did not outperform "SVM-iWUM (α=1)" for two categories, when we take its above-mentioned advantageous conditions into consideration, we can reasonably conclude that our method performed rather well and succeeded in exploiting information in the surrounding pages by

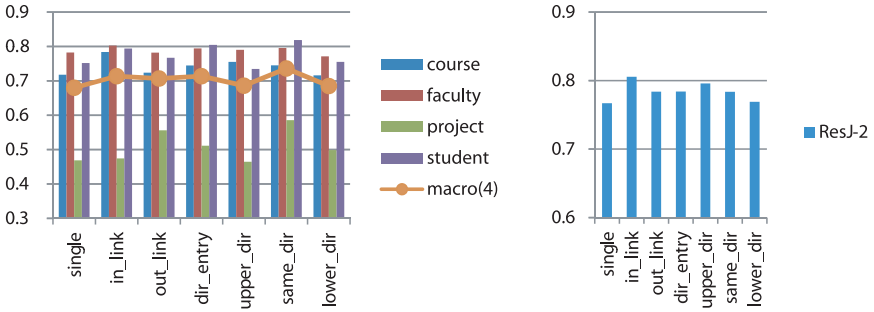


Fig. 6 Contribution of element SPGs to basic classification performance.

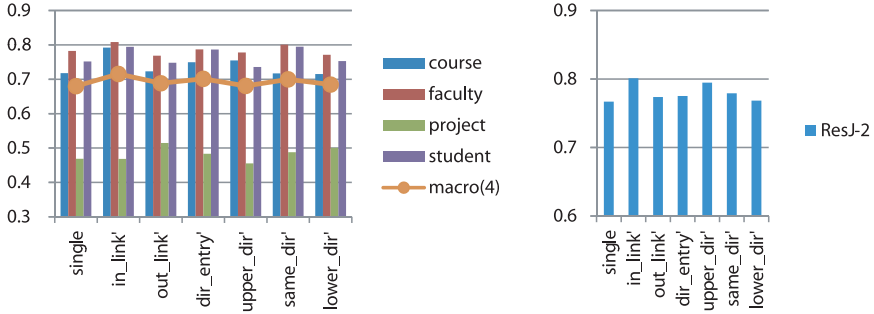


Fig. 7 Contribution of merged SPGs to basic classification performance.

using a relatively simple process. Moreover, our method was the most stable one in all four categories.

Experiment 2: Contribution of element SPGs

To find out which parts of the surrounding pages contribute to the performance improvement of $\mathcal{M}_{\text{full}}$, we tested the following supplemental SP models:

$$\mathcal{M}_h = \{S_{\text{entry\_page}}\} \cup \bigcup_{i \in R_c} \{S_{i,h}\}$$

$$\mathcal{M}_c = \{S_{\text{entry\_page}}\} \cup \bigcup_{j \in R_h} \{S_{c,j}\}$$

for each $h \in R_h, c \in R_c$, and compared the results with those of $\mathcal{M}_{\text{single}}$ (see **Fig. 6**).

For both Web-KB and ResJ-2, all the tested SP models are superior or equivalent to $\mathcal{M}_{\text{single}}$. This implies that all element SPGs are effective (or at least harmless).

The results vary among SP models and data sets. For instance, $\mathcal{M}_{\text{same\_dir}}$ is more effective for Web-KB than for ResJ-2, whereas $\mathcal{M}_{\text{in\_link}}$ is the reverse. It is remarkable that $\mathcal{M}_{\text{upper\_dir}}$ is much more effective on ResJ-2 but almost ineffective on Web-KB. We believe the reason is that since Web-KB consists of a small number of web sites and for each university, the majority of the pages share the directory path, as mentioned in Section 4.1, and consequently, the classifier could not find effective characteristics from those features. No clear tendency can be found across the categories of the Web-KB data set; some SP models work well for some categories, but not for others.

Experiment 3: Comparing contributions of merged SPGs

We tested SP models that were similar to those used in Experiment 2, but using SPGs made by merging corresponding element SPGs:

$$\mathcal{M}_h\prime = \{S_{\text{entry\_page}}\} \cup \left\{ \bigcup_{i \in R_c} S_{i,h} \right\}$$

$$\mathcal{M}_c\prime = \{S_{\text{entry\_page}}\} \cup \left\{ \bigcup_{j \in R_h} S_{c,j} \right\},$$

and obtained the results shown in **Fig. 7**.

For both Web-KB and ResJ-2, all the tested SP models are superior or equivalent to $\mathcal{M}_{\text{single}}$. This means that all SPGs are effective (or at least harmless), even though they are merged.

The results of experiment 3 vary among SP models and data sets. Comparing the overall performances in experiment 3 with those of experiment 2, we see that the merged SPGs of experiment 3 lead to slightly smaller improvements from $\mathcal{M}_{\text{single}}$ for both ResJ-2 and Web-KB. Upon checking the corresponding SP models in experiments 2 and 3 one by one, we can see that experiment 3 gave better performance only in a small number of cases. Therefore, we may conclude that using element SPGs separately is more effective than using merged SPGs on average.

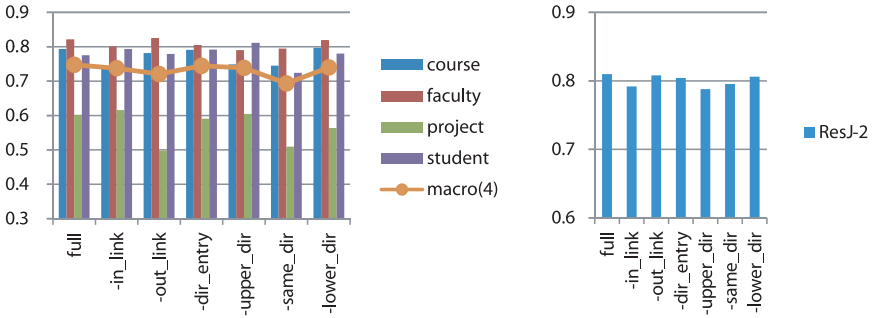


Fig. 8 Effect of excluding element SPGs from full-structured SP model.

Experiment 4: Effect of excluding element SPGs

To see the effect of using multiple SPGs from a different aspect, we tested SP models excluding the element SPGs of each relation from $\mathcal{M}_{\text{full}}$:

$$\mathcal{M}_{-h} = \mathcal{M}_{full} \cap \overline{\bigcup_{i \in R_c} \{S_{i,h}\}}$$

$$\mathcal{M}_{-c} = \mathcal{M}_{full} \cap \overline{\bigcup_{j \in R_h} \{S_{c,j}\}},$$

and obtained the results shown in **Fig. 8**. The effect of the excluded element SPGs can be checked by looking at the differences from $\mathcal{M}_{\text{full}}$; i.e., a larger difference (lower performance) means a larger contribution.

For both Web-KB and ResJ-2, all the tested SP models are inferior or equivalent to $\mathcal{M}_{\text{full}}$. This implies again that all element SPGs are effective (or at least harmless).

Now let us compare the performance losses from $\mathcal{M}_{\text{full}}$ with the performance gains from $\mathcal{M}_{\text{single}}$ in experiment 2. They correspond very well except for $\mathcal{M}_{\text{-dir\_entry}}$ for Web-KB. Their absolute values for Web-KB were of the same level, whereas those for ResJ-2 were two to three times smaller in comparison with experiment 2. This means that, the redundancies of the element SPGs were small for Web-KB, but were rather large for ResJ-2; hence, removing some element SPGs did not have a large effect for the latter case.

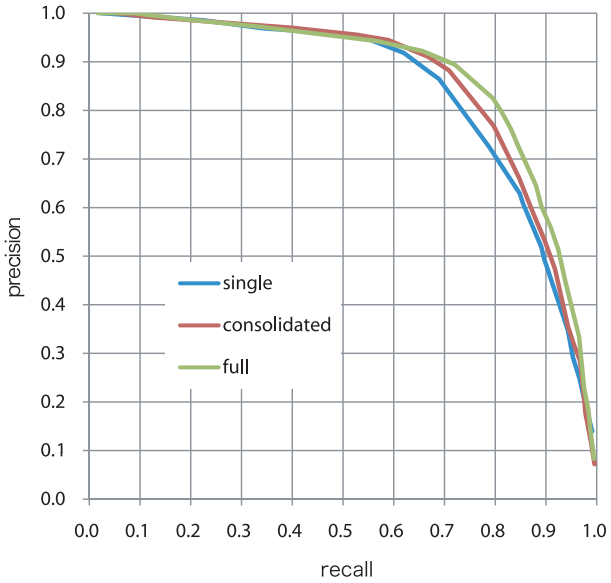


Fig. 9 Performance in all recall-precision ranges.

4.3 Experiments on Quality Controlled Performance

We tested our method under the conditions of high recall or high precision. We used only the ResJ-2 data set for the following reason: As described in Section 4.1, the *leave one university out* cross validation method for the Web-KB data set seems to have unpredictable side effects. Since EP's context in the directory structure is rather uniform for the test data of each fold, the features from the upper directories may cause over-fitting and the behavior of the classifier will consequently be unstable.

We used $\mathcal{M}_{\text{full}}$ in the following experiments. $\mathcal{M}_{\text{single}}$ and $\mathcal{M}_{\text{consolidated}}$ were used for comparison.

Experiment 5: Performance in broader precision-recall range

Figure 9 plots the performance of $\mathcal{M}_{\text{single}}$, $\mathcal{M}_{\text{consolidated}}$, and $\mathcal{M}_{\text{full}}$ in all recall-precision ranges. We tested various c and j option parameters of SVM^{light} and drew each curve by connecting the results of the well-performing ones. $\mathcal{M}_{\text{full}}$ significantly outperforms $\mathcal{M}_{\text{single}}$ and $\mathcal{M}_{\text{consolidated}}$ from the medium to high

Table 2 Performance of well performing classifier composition.

SP model	F-measure	Recall at precision p			Precision at recall r		
		p =.98	p =.99	p =.995	r =.90	r =.95	r =.98
$\mathcal{M}_{\text{single}}$	.767	.263	.160	.092	.483	.308	.185
$\mathcal{M}_{\text{consolidated}}$	.786	.272	.146	.096	.525	.334	.164
$\mathcal{M}_{\text{full}}$	.810	.259	.165	.126	.581	.398	.202

recall range (0.6–1.0). However, it did not show any gain at high precision (0.95–1.0). $\mathcal{M}_{\text{consolidated}}$ is a little better than $\mathcal{M}_{\text{single}}$ in the middle recall-precision range. This result confirms that using the surrounding pages with our method enables us to collect relevant web pages comprehensively, although they make only a small contribution when they are bundled together.

Experiment 6: Performance with controlled precision/recall

We need to obtain two component classifiers that perform possibly the best at a given precision and a given recall to compose the three-grade classifier. We can tune them independently with the c and j option parameters of SVM^{light} under the respective constraints. The second column in **Table 2** shows the F-measures corresponding to the plots in the right graph of Fig. 4, the third to fifth columns show the recall values at the precisions given in the second row, and the sixth to eighth columns show the precision values at the recalls given in the second row. Note that since it was impossible to obtain the exact precisions and recalls given in the second row by tuning the parameters, the values in the third to fifth rows were interpolated from the surrounding values.

The recalls at high precisions seem to have no meaningful differences. Moreover, as seen in Fig. 9, they are affected by small variations in precision that may be caused by a small number of false positives.

On the other hand, the proposed method clearly improved the precisions at high recalls. Although the precisions are also affected by small variations in the recall, the effect of false negatives is relatively small. A T-test showed that the precisions of $\mathcal{M}_{\text{full}}$ and $\mathcal{M}_{\text{single}}$ at each recall were significantly different ($p < 0.05$).

5. Considerations

5.1 Effectiveness of Proposed Features

(1) Overall result

The effectiveness of our method on the basic performance was verified in experiment 1. Although the effect of each SPG depended on the data set and the category, all of them had positive effects to some degree for some cases, and almost no negative effects were observed. Therefore, we can expect $\mathcal{M}_{\text{full}}$ to perform nearly the best, although we may be able to achieve better performance by selecting only the effective SPGs according to the data sets and categories through experimentation.

Considering that $\mathcal{M}_{\text{consolidated}}$ improves performance only to a limited degree, even though it used the same textual information as $\mathcal{M}_{\text{full}}$, we suppose that the semantic relation of the surrounding pages can be properly represented with our method and consequently that the effect of noisy information from the surrounding pages can be suppressed to some degree.

(2) Results for Web-KB

Comparing our method, the baseline, and the previous methods for each category of the Web-KB data set, we can see that our method performed stably even when the training data size was small, as in the case of the *project* category. This is an especially big advantage for dealing with real-world problems.

As seen in Fig. 4 and Table 1, the performance improvement for the *project* category was the largest among the four categories. The performance gain for "macro(4)" was 0.067 (= 0.739 − 0.672), of which 0.029 (= (0.569 − 0.454)/4, approximately 43%) comes from this category. Since the baseline performance was low, the information in the EPs of this category was probably insufficient for classification, while the performance gain was achieved by using the information from the SPs. The baseline performances for the other categories were rather high, implying that each EP included sufficient information for web page classification. Therefore, we looked more closely at the results for the *project* category.

Before discussing the contribution of SPGs, we describe what we observed by checking the data in the *project* category. The majority of the EPs were in each

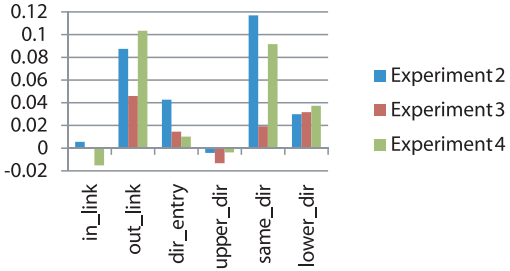


Fig. 10 Contribution of SPGs for the *project* category.

user's web home directory or in its child directory. Taking into account that no sibling or collateral pages are included in SPs, the EPs usually have almost no in-links from the SPs other than the user's own pages. On the other hand, the EPs often have several component pages in the same or lower directories. In addition, the user's home page is usually included in the SPs of the dir_entry relation (if it is the user's web home directory), as well as in the SPs of the out_link relation.

Figure 10 shows the differences between the result of each SP models and those of $\mathcal{M}_{\text{single}}$ in Fig. 6 and Fig. 7 and $\mathcal{M}_{\text{full}}$ in Fig. 8. Each value indicates the contributions of respective SPGs. Negative values indicate negative effects.

From the results of experiments 2 through 4, we can say that SPGs of the same_dir and out_link relations significantly contributed to improving classification performance, while those of the lower_dir and dir_entry relations contributed to a lesser degree. It is interesting that those of the same_dir were not so effective only for experiment 3. These results can be explained as follows taking the above-mentioned observations on the data into consideration.

The same directory pages frequently contained component pages and the user's home page, which were quite informative, as well as other miscellaneous pages (probably as in-link pages). They were best utilized when used as separate element SPGs. However, when the element SPGs were bundled together, the component pages were mixed with some noisy pages.

The out-link pages also frequently contained component pages and the user's home page, which were quite informative, and sometimes the department's top pages, which were also informative. Since these pages had fewer noisy pages than in the same directory pages, they were effective to some degree even when the element SPGs were bundled together.

In contrast to the above results, the SPGs of the upper_dir and in_link relations tended to have no effect or rather negative effects. For upper_dir, the tendency was found in all the four categories, and this was probably due to the characteristic of the data set that the web pages were crawled from small number of web servers. For in_link, the tendency was observed only in the *project* category, and these SPGs were rather effective for the other categories. This reason is considered to be as follows. As mentioned above in the observation, there were almost no in-links from the pages of the organization (e.g., university, department). Moreover, a project EP usually has in-links from almost all of its descendant pages. Consequently, since the in-link pages were very diverse, the sample size was too small to learn any regularity from those pages.

(3) Results for ResJ-2

As presented in Fig. 9, our method was effective at improving precision especially in the high recall range. In light of some of the other experimental results not shown here, the features of the element SPGs of the in_link and of upper_dir relations contributed significantly. This probably indicates that the SP that had a hyperlink to the EP or that were placed in the upper directory provided contextual information that was lacking in the EP itself.

On the other hand, the contributions of the features from the other element SPGs were below our expectations. This is probably because there are not many target pages whose actual information is contained only in their component pages. However, a certain number of such pages actually exist and can never be collected without the features from those pages.

5.2 Application to Three-grade Classifier

To determine the effectiveness of applying our method to a three-grade classifier, we evaluated the reduction of the uncertain page amount. We compared two compositions of the three-grade classifiers: one using $\mathcal{M}_{\text{single}}$ and the other using $\mathcal{M}_{\text{full}}$ for both the recall-controlled and precision-controlled classifiers.

Three-grade classification is executed in the following way: (1) All data are classified by the recall-controlled classifier, and its negative output data are labeled as "assured negative"; (2) The remaining data are further classified by

Table 3 Estimated page number of classification output from the NW100G-01 data set.

Controlled performance (precision/recall)	$\mathcal{M}_{\text{single}}$			$\mathcal{M}_{\text{full}}$			Reduction ratio $\lvert U_{\text{full}}\rvert/\lvert U_{\text{single}}\rvert$
	assured positive	uncertain U_{single}	assured negative	assured positive	uncertain U_{full}	assured negative	
.995/.98	4,415	250,294	10,784,011	6,093	227,109	10,805,518	0.907
.99/.95	7,753	140,268	10,890,699	8,014	106,633	10,924,073	0.760
.98/.90	12,867	76,513	10,949,341	12,691	61,677	10,964,352	0.806

the precision-controlled classifier, and its positive and negative output data are labeled as "assured positive" and "uncertain", respectively.

Table 3 shows the estimated numbers of the pages in the output classes and the reduction ratios of the uncertain class size for three different quality requirements when the three-grade classifier is applied to the whole NW100G-01 data set.

We estimated the numbers in the following way. We prepared ResJ-2 using the following process: (1) filter NW100G-01 with a method presented in Ref. 14), (2) sample 1%, and (3) assess manually. Therefore, the number of positive data in the whole corpus is calculated by using $|P_{\text{corpus}}| = |\text{positive\_sample}|/\text{sampling\_rate}$, ignoring the false negative in step (1). The size of the assured positive class is calculated as $|P_P|$ by transforming the definition formulas of the precision and recall under the controlled precision and substituting P_{corpus} for P_T. The size of the assured positive class and the uncertain class together is calculated in the same way as mentioned above under the controlled recall. Consequently, the sizes of the classification results of the three-grade classifier are calculated using the following equations:

$$|\text{assured\_positive}| = |P_{\text{corpus}}| \times \frac{\text{recall\_at\_precision\_given}}{\text{precision\_given}}$$

$$|\text{uncertain}| = |P_{\text{corpus}}| \times \frac{\text{recall\_given}}{\text{precision\_at\_recall\_given}} - |\text{assured\_positive}|$$

$$|\text{assured\_negative}| = |\text{corpus}| - |\text{assured\_positive}| - |\text{uncertain}|.$$

The results show that, compared with $\mathcal{M}_{\text{single}}$, $\mathcal{M}_{\text{full}}$ evidently decreased the uncertain class size, although no increase of the assured positive class size is apparent.

We should note that the uncertain class size is affected by the assured negative class size, whereas the assured positive class size has an almost no effect, because the absolute size of the former is larger than that of the latter by 2 to 3 orders. Since the recall-precision plot is very steep in the very high recall area, a small performance difference makes a large difference in the assured negative class size. Therefore, for the specific purpose of reducing the uncertain class size, improving the performance of the recall-controlled classifier, even by a small margin, is a crucial task. In this regard, our method is effective for building web page collections.

6. Conclusion and Future Work

We proposed a web page classification method in which we introduced the idea of using a page group model for generating a feature vector from local surrounding pages. We demonstrated through experimentation the effectiveness of extracting textual features from separate element SPGs taking into consideration the connection type and the directory hierarchy; the method is effective not only for the basic performance around the breakeven point, but also for the quality-controlled performance in the high recall range. Consequently, the proposed method is effective in reducing the uncertain class size when it is used in a three-grade classifier.

However, the classification performance is still lower than a human's ability. Considering how people use the surrounding pages for assessing the entry page, it will be essential to discriminate the surrounding pages that compensate for the lacking information in the entry page. Therefore, in the future, we will investigate ways to estimate the usefulness of the surrounding pages and will incorporate them into the current scheme.

Nevertheless, since the proposed method is simple and easy to implement, it can be used together with various existing classification techniques and features from other various information sources such as URL patterns, anchor texts, and page structures, which have been shown to be effective in previous studies.

We tested our method with two data sets whose categories are all from the academic field. Thus, its applicability to other fields has yet to be proven. However, we assume it is effective for a wide range of categories, because the link and directory structures it uses are common to various web sites and can be used with-

out additional training data or domain-specific knowledge. For instance, pages in corporate web sites, e.g., employment information and product guide pages, have rather common information types and conceptual structures, although web page structures vary according to the size and domain of the company, and our method is considered to be effective enough for use with them. Note though that since article pages on blogs are created sequentially, their link structure is rather different from ordinary web sites, and almost no semantic relations are reflected in the directory structure. Therefore, our method cannot be applied to them without modification. On the other hand, the top pages of blogs are a good target for our method because they have common characteristic component pages, such as personal profiles and article index pages.

In addition, we need to resolve the processing cost problems because high-performance classifiers require rather complex feature processing, whereas the real size of the Web is enormous. We have partly tackled this problem by using the rough filtering presented in Ref. 14) and expect to be able to overcome it by extending our approach.

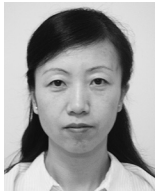
Acknowledgments This research was partially supported by a Grant-in-Aid for Scientific Research B (No.18300037) from the Japan Society for the Promotion of Science (JSPS). We used the NW100G-01 document data set under permission from the National Institute of Informatics (NII). We would like to thank Professors Akiko Aizawa and Atsuhiro Takasu of NII for their helpful advice, and the editor in charge and the anonymous reviewers for their helpful comments on improving the paper.

References

1) Chakrabarti, S.: Data mining for hypertext: A tutorial survey, *ACM SIGKDD Explorations*, Vol.1, No.2, pp.1–11 (2000).
2) Sun, A., Lim, E.-P. and Ng, W.-K.: Web classification using support vector machine, *Proc. 4th international workshop on web information and data management*, McLean, Virginia, USA, pp.96–99, ACM Press (2002).
3) Craven, M. and Slattery, S.: Relational Learning with Statistical Predicate Invention: Better Models for Hypertext, *Machine Learning*, Vol.43, No.1-2, pp.97–119 (2001).
4) Sun, J., Zhang, B., Chen, Z., Lu, Y., Shi, C. and Ma, W.: GE-CKO: A Method to Optimize Composite Kernels for Web Page Classification, *Proc. 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (*WI2004*), Beijing, China, pp.299–306 (2004).
5) Shih, L.K. and Karger, D.R.: Using URLs and table layout for web classification tasks, *Proc. WWW2004*, New York, NY, USA, pp.193–202 (2004).
6) Kan, M.-Y. and Thi, H.: Fast webpage classification using URL features, *Proc. CIKM'05*, Bremen, Germany, pp.325–326 (2005).
7) Kan, M.-Y.: Web Page Categorization without the Web Page, *Proc. 13th World Wide Web Conference* (*WWW2004*), New York, NY, USA (2004).
8) Masada, T., Takasu, A. and Adachi, J.: Improving web search performance with hyperlink information, *IPSJ Transactions on Databases*, Vol.46, No.8, pp.48–59 (2005).
9) Sun, A. and Lim, E.-P.: Web unit mining: finding and classifying subgraphs of web pages, *Proc. International Conference on Information and Knowledge Management* (*CIKM2003*), New Orleans, Louisiana, USA, pp.108–115 (2003).
10) Yang, Y., Slattery, S. and Ghani, R.: A Study of Approaches to Hypertext Categorization, *Intelligent Information Systems*, Vol.18, pp.219–241 (2002).
11) Chakrabarti, S., Dom, B. and Indyk, P.: Enhanced hypertext categorization using hyperlinks, *Proc. Int. Conf. Management of Data* (*SIGMOD '98*), Seattle, WA, USA, pp.307–318 (1998).
12) Chau, M.: Applying web analysis in web page filtering, *Proc. ACM/IEEE-CS Joint Conference on Digital Libraries* (*JCDL'04*), Tucson, Arizona, USA, p.376 (2004).
13) Glover, E.J., Tsioutsiouliklis, K., Lawrence, S., Pen-nock, D.M. and Flake, G.W.: Using web structure for classifying and describing web pages, *Proc. 11th International World Wide Web Conference*, Honolulu, Hawaii, USA, pp.562–569 (2002).
14) Wang, Y. and Oyama, K.: Combining page group structure and content for roughly filtering researchers' homepages with high recall, *IPSJ Transactions on Databases* (*IPSJ TOD*), Vol.47, No.SIG 8, pp.11–23 (2006).
15) Cover, T.M. and Thomas, J.A.: *Elements of information theory*, Wiley (1991).
16) Eguchi, K., Oyama, K., Ishida, E., Kando, N. and Kuriyama, K.: Overview of the web retrieval task at the third NTCIR Workshop, NII Technical Report NII-2003-002E, National Institute of Informatics (2003).
17) Eguchi, K., Oyama, K., Ishida, E., Kando, N. and Kuriyama, K.: Evaluation methods for web retrieval tasks considering hyperlink structure, *IEICE Trans. Inf. Syst.*, Vol.E86-D, No.9, pp.1804–1813 (2003).

(Received December 20, 2008)
(Accepted April 8, 2009)

(Editor in Charge: *Toshiyuki Amagasa*)



Yuxin Wang received her B.E. and M.E. from East China Normal University, Shanghai, China in 1990 and 1993, respectively, and Ph.D. from the Graduate University of Advanced Studies (SOKENDAI), Japan in 2006. She became a project researcher at the National Institute of Informatics (NII) in 2006, a research fellow at the University of Tokyo in 2007, and joined Team Lab Corp. in 2008. Her research interests include web information utilization, information retrieval, and natural language processing.

Keizo Oyama received his B.E., M.E., and Dr. Eng. from the University of Tokyo in 1980, 1982, and 1985, respectively. He is a professor at the National Institute of Informatics (NII) and the Graduate University for Advanced Studies (SOKENDAI). His research interests include web information access technology, full-text information retrieval, and structured text processing. He is a member of IPSJ, IEICE, JSIMS, and DBSJ.