Classifying web content using discriminant efficiencies

Akira Maeda[1], Yukinori Hayashi[2]
[1]College of Information Science and Engineering
Ritsumeikan University
1-1-1 Noji-higashi, Kusatsu
Shiga 525-8577, Japan
amaeda@is.ritsumei.ac.jp

[2]EXA Corporation
Solid Square East Tower, 580 Horikawa-cho
Saiwai-ku, Kawasaki,
Kanagawa 212-8555, Japan
lure00@gmail.com

ABSTRACT: *In this paper, we propose a method to classify Web documents by genre (not by topic) based on features of terms and HTML tags. For the classifier, we use SVM (Support Vector Machine) and Naïve Bayes. In order to improve the accuracy of classification, we calculate discriminant efficiencies of each pair of a term and a HTML tag to find out HTML tags which are effective for genre classification. We conducted experiments of genre classification of Japanese Web documents using the proposed method. The experimental results show that our method using discriminant efficiencies achieves 8% increase in classification accuracy.*

Keywords: Genre classification, Discriminant efficiency, SVM, Naïve Bayes

Received: 18 April 2010, Revised 21 May 2010, Accepted 28 May 2010

1. Introduction

Rapid growth of the Web has increased the possibility that information which the user is looking for exists, but at the same time it has made it difficult to get to such information from enormous amount of Web documents.

In order to tackle this problem, various techniques have been proposed, including recommendation engines and personalized search. One of such techniques is automatic classification of Web documents. It is often combined with Web search engines, in a way that the Web documents in the search results are classified into predefined or dynamically generated classes. Examples of such search engines include Clusty[1]and WebClust[2]. For the classification of Web documents, traditional text classification or clustering techniques can be used.

However, most of such techniques concentrate on classification based on topic of a document, but not on document genre. By "document genre" we refer to the type of a Web document, such as a corporate site, a news site, an online shopping site, and CGM (Consumer Generated Media) such as blogs.

In this paper, we propose a genre classification method which can be used to classify the search results of Web search engines. Our proposed method is based on standard classification techniques, namely SVM (Support Vector Machine) and Naïve Bayes. In order to increase the accuracy of classification, we calculate discriminant efficiencies for finding out useful features for genre classification.

The rest of the paper is organized as follows: Section 2 gives a brief explanation of genre classification. Section 3 reviews related work. Section 4 describes our proposed method in detail. Section 5 presents the results of the experiments. Section 6 discusses the experimental results in detail. Finally, Section 7 concludes the paper.

2. Genre classification

In traditional topic-based classification, documents are classified by topics, such as politics, sports, economics, etc. Let us consider an example of “iPod” as a topic. The documents related to “iPod” have many different aspects such as the official site, online shopping sites of iPod, blog entries that review impressions of iPod, news stories related to iPod, etc.

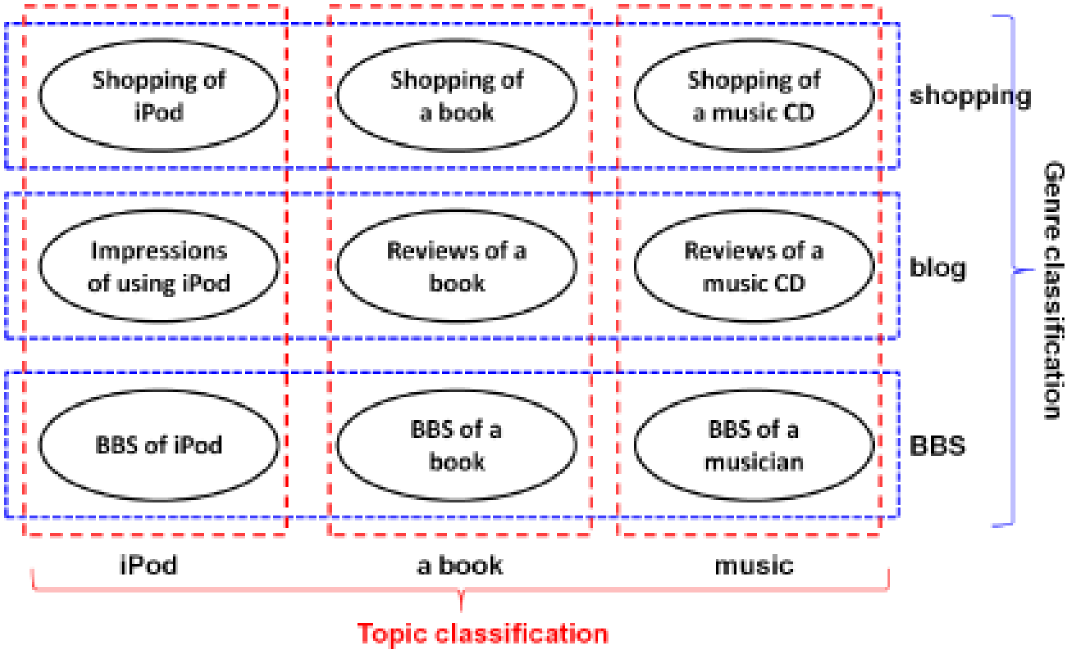

Figure 1. Difference between topic classification and genre classification

Thus, if we input “iPod” as the query for a Web search engine, the user has to look for the documents of desired genre from a long list of search results. If the user can think of some additional query terms to narrow down the document genre, it will make it easier to find the desired documents. However, it is often difficult to think of such additional query terms, and sometimes there is no term which can effectively narrow down the document genre. For example, it is difficult to search only reviews of iPod by adding query terms.

Genre classification technique can be a solution to these problems. Shows the difference between topic classification and genre classification. If the user inputs only “iPod” as the search term, the system can classify the search results into some predefined genres, such as the official sites, online shopping, review, news, etc.

3. Related work

Genre classification of Web documents has been extensively studied in recent years. It is a challenging task, because it is difficult to find appropriate features and to extract them from the Web pages. For the features to be used for genre classification, HTML tags and URLs are most commonly used in many studies.

Vidulin et al.[2] uses the detailed analysis of URL features which consists of 76 features. Chaker et al.[3] uses URL for contextual classifier and HTML tags for structural classifier. Levering et al.[4] uses textual features, HTML features, and

[1] *http://clusty.com/*
[2] *http://www.webclust.com/*

visual features that capture the layout characteristics of the genres, and show that visual features are useful for fine-grained genre classification.

Dong et al.[5] uses three attributes for genre classification; Content, Form, and Functionality. Content and Form are based on textual and HTML tag features, respectively. Functionality is also based on HTML tags, but it includes navigation links, Java applet, JavaScript, JSP, etc.

Ferizis et al.[6] conducted performance analysis of genre classification of Web documents and proposed a method to improve the efficiency of linguistic analysis.

Santini[7] analyzed the effects of corpus composition for learning, genre palette (categories), and feature representativeness in genre classification, and showed the limitations on the exportability of classification models to a different collection. They also discuss about the limitations of a single-label discrete classification strategy for genre classification of Web pages.

4. The proposed method

4.1 Overview of the proposed system

As described in Section 1, our aim is to propose an automatic genre classification method which can be used to classify the search results of Web search engines. In this section, we explain the overview of the whole system which integrates the proposed method, Web search API, and user interface.

Shows the overview of the system. First, a user inputs a query to the system (1), and the system passes the query to a Web search API, such as Yahoo! Web Search API (2). Then, the system receives the search results from the API as a list of URLs (3). The system then downloads each Web document in the search result list (4 and 5). For each document downloaded, the system classifies it in terms of document genre using our proposed method (6). Finally, the results are returned to the user (7).

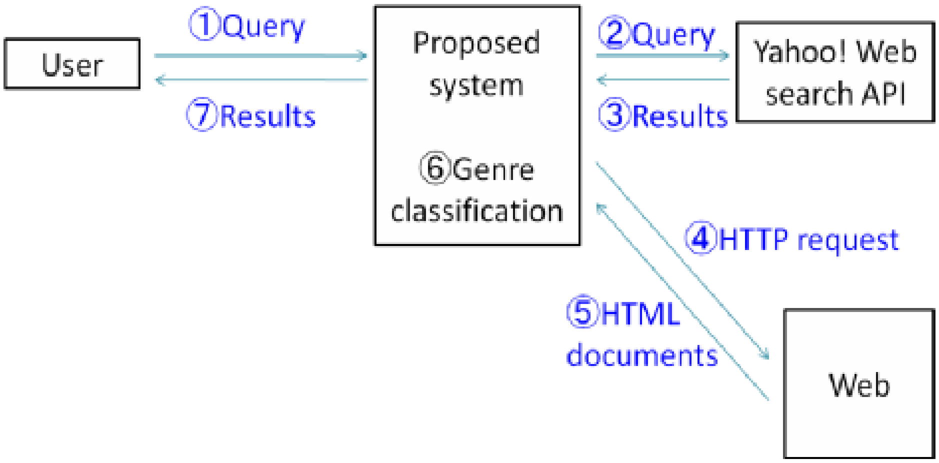

Figure 2. Overview of the proposed system

4.2 Document Vectors

In the genre classification phase (6).We first create a feature vector for each document. Feature vector is based on term frequency. Dimensions of the vector are distinct terms in the document corpus, and the value of each element is the occurrence frequency of that term. As we explain in the next section, the "term" may be a pair of an HTML tag and a term, or an HTML tag alone. Shows the example of making the document vector from a Web document.

4.3 Using HTML tags

In order to classify Web documents by genre, we use several HTML tags which are considered to be useful for genre classification. For selecting useful HTML tags, we first conducted experiments of calculating discriminant efficiencies of each term and each term/tag pair, and manually selected some useful tags. Calculation of discriminant efficiency is explained in section 4.5, and the experimental results are described in section 5.2. The selected tags based on the experiments are <title>, <a>, <form>, and <input>.

For <title> tag, we count the terms which are surrounded by <title> tag with tag information. For example, a document fragment "<title>Foo bar baz</title>" is broken into "<title>Foo", "<title>bar", and "<title>baz".

For <a> tag, in addition to the count method for <title> tag, we distinguish between internal links and external links. Besides, we count the whole number of links and the number of internal links.

For <form> tag, we count the frequency of <form> tags in the document. In this case, we only use the frequency of the tag, and do not consider terms within that tag.

For <input> tag, we count the frequency of the tags in the same way as <form> tag. In addition, if its "type" attribute is "submit", we extract terms from "value" attribute. For example, a document fragment "<input type="submit" value="foo bar">" is broken into "<submit>foo" and "<submit>bar".

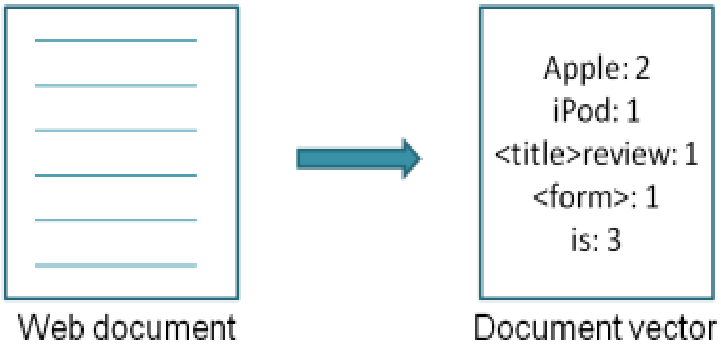

Figure 3. Example of making the document vector from a Web document

4.4 Classification method

For the classifier, we use SVM (Support Vector Machine) and Naïve Bayes. SVM is a widely used supervised machine learning method which is shown to be effective for classification task. Naïve Bayes is a simple probabilistic classifier based on Bayes theorem. In the experiments, we compare the results of these two methods used for our proposed genre classification approach.

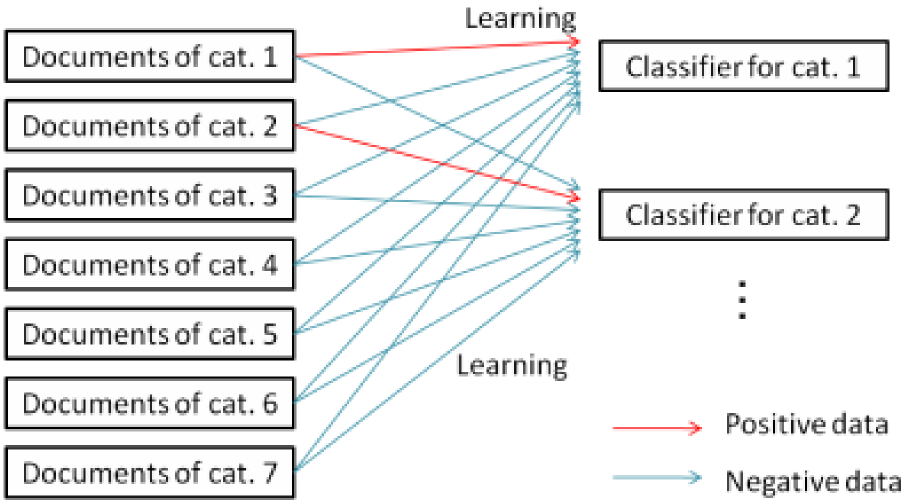

Figure 4. The method of making a classifier for each category

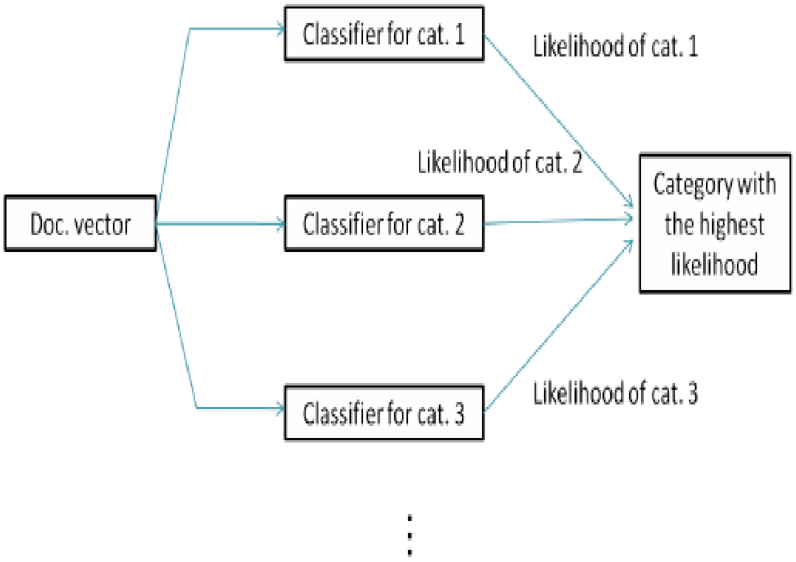

Figure 5. The method of classifying a document using multiple classifiers

Since SVM and Naïve Bayes are binary classifiers, we have to extend them into multi-class classifiers. For this, we use standard “one vs. rest” method. For training, we train classifiers by using the documents which fall into that class as positive, and the rest of the documents as negative. Figure 4 and 5 illustrate the procedure of “one vs. rest” method.

4.5 Discriminant analysis

Discriminant analysis is a method to estimate which population a sample belongs to from multiple populations. It is calculated based on mean and variance of populations, and Mahalanobis distance between a sample and populations.

Given two populations *P* and *Q*, means and variances of these populations can be calculated by the following formulas:

Mean of population *P*:

$$\mu_P = \frac{1}{n_P} \sum_{i=1}^{n_P} x_{Pi} \qquad (1)$$

Mean of population *Q*:

$$\mu_Q = \frac{1}{n_Q} \sum_{i=1}^{n_Q} x_{Qi} \qquad (2)$$

Variance of population *P*:

$$\sigma_P^2 = \frac{1}{n_P - 1} \sum_{i=1}^{n_P} (x_{Pi} - \mu_P)^2 \qquad (3)$$

Variance of population *Q*:

$$\sigma_Q^2 = \frac{1}{n_Q - 1} \sum_{i=1}^{n_Q} (x_{Qi} - \mu_Q)^2 \qquad (4)$$

where n_P and n_Q are the numbers of samples in the populations P and Q, respectively, and x_{Pi} and x_{Qi} are the samples in the population P and Q, respectively.

Given a sample x, in order to estimate which population the sample belongs to, we can define the square of Mahalanobis distance between the sample and each population as follows:

$$D_P^2 = \frac{(x - \mu_P)^2}{\sigma_P^2} \quad (5)$$

$$D_Q^2 = \frac{(x - \mu_Q)^2}{\sigma_Q^2} \quad (6)$$

One way of discriminant analysis is to compare these distances and estimates that sample x belongs to the population with larger distance.

Discriminant efficiency is the square of Mahalanobis distance between two population means, which increases with the probability of errors decreases. Thus, higher discriminant efficiency means more accurate classification. The discriminant efficiency of two populations P and Q can be calculated as follows:

$$D^2(P,Q) = \frac{(\mu_P - \mu_Q)^2}{\sigma^2} \quad (7)$$

5. Experiments

In order to verify the effectiveness of our proposed method, we conducted experiments of genre classification. Firstly, we calculated discriminant efficiencies for each term extracted from the documents. Then, we classified the documents based on the results of the discriminant efficiencies.

5.1Categories to be classified

In this experiment, we defined 7 categories as follows: “news”, “shopping”, “internet service”, “diary” (which includes blogs), “bbs” (bulletin board system), “portal”, and “commentary”. For each category, we manually collected appropriate Web pages from the Internet. Note that we used only the Web pages written in Japanese. However, our proposed method does not depend on the language and can easily be adopted for other languages. The number of documents in each category is shown in Table 1.

Category	No. of docs
news	120
shopping	132
internet service	100
diary	217
bbs	145
portal	57
commentary	99

Table 1. Number of documents in each category

5.2 Calculation of discriminant efficiencies

For calculating discriminant efficiencies, we compared the result of one particular category and the sum of the results for the rest of the categories. For example, the discriminant efficiency is calculated between “diary” and the rest of 6 categories. Since discriminant efficiencies show the degree to which the averages of elements in two sets are separated, in this case, both the terms which frequently appear in “diary” and rarely appear in the rest of the categories, and the terms that rarely appear in “diary” and frequently appear in the rest of the categories, will have higher discriminant efficiencies.

Firstly, we calculated the discriminant efficiencies for the frequencies of the terms without HTML tags, in the same way as the case of normal text classification. The top 10 terms that have high discriminant efficiencies in “commentary” category are shown in Table 2.

Rank	Discriminant Efficiency	Term (English translation)
1	0.00376	(.) (foundation)
2	0.00279	..(ministry of finance)
3	0.00279	after
4	0.00247	.... (to describe)
5	0.00247	.. (defeat)
6	0.00184	.. (exile)
7	0.00184	.. (shogun’s harem)
8	0.00184	. (cloudiness)
9	0.00184	.
10	0.00184	...... (superheat)

Table 2. Top 10 terms for “commentary” category

As illustrated in Table 2, text classification with very high dimensions results in very low discriminant efficiencies that are not so useful for accurate classification. Thus, we need “terms” with higher discriminant efficiencies. In order to achieve this, we use HTML tags along with the terms.

Table 3 and 4 show the top 10 terms within <TITLE> tags that have high discriminant efficiencies in “shopping” and “internet service” categories, respectively.

Show the top 10 terms within <A> tags that have high discriminant efficiencies in “news” and “shopping” categories, respectively.

Show the top 10 terms within <FORM> and <INPUT> tags that have high discriminant efficiencies in “shopping” and “bbs” categories, respectively.

As illustrated from Tables 3 to 8, using HTML tags along with the terms themselves greatly improves “informativeness” of the terms with high discriminant efficiencies. Based on these observations, we selected tags <title>, <a>, <form>, and <input> along with the terms themselves as the features for the classification experiments described in the next section.

5.3 Experiments of classification accuracy

Based on the observation of the experiments in the previous section, we selected <TITLE>, <A>, <INPUT>, and <FORM> tags along with the term frequencies. The details of how we used HTML tags and term frequencies as the features are explained in section 4.3. For the experiments of classification accuracy, we conducted 5-fold cross-validation using these features. We compared the classification accuracies using SVM (Support Vector Machine) and Naïve Bayes with the same data set. The results of the experiments are shown.

6. Discussion

From the observation of the experimental results in section 5.3, we can see that the improvements by using HTML tags are significant for SVM, but not for Naïve Bayes. In classification using Naïve Bayes, the terms with higher weights are mostly

Rank	Discriminant Efficiency	Term (English translation)
1	0.00839	.. (mail order)
2	0.00436	...... (shopping)
3	0.00410	..... (Bidders; Japanese auction site)
4	0.00321	.. (sale)
5	0.00304	.... (shop)
6	0.00280	Amazon
7	0.00238	..... (channel)
8	0.00238	.. (postage)
9	0.00197	.... (catalog)
10	0.00197	GDOSHOP

Table 3. Top 10 terms within <TITLE> tags for "shopping" category

Rank	Discriminant Efficiency	Term (English translation)
1	0.00270	.... (server)
2	0.00258	.. (free)
3	0.00239	.... (domain)
4	0.00229	.... (rental)
5	0.00225	... (mail)
6	0.00204	.. (map)
7	0.00178	... (Wiki)
8	0.00178	.... (affiliate)
9	0.00178	..... (keywords)
10	0.00178	infoseek

Table 4. Top 10 terms within <TITLE> tags for "internet service" category

Rank	Discriminant Efficiency	Term (English translation)
1	0.00954	.. (international)
2	0.00707	.. (continued drop)
3	0.00693	.. (subscription)
4	0.00629	.. (scandal)
5	0.00618	. (punctuation mark)
6	0.00617	.... (top news)
7	0.00613	.. (great, extensive)
8	0.00609	.... (news)
9	0.00609	.. (politics)
10	0.00603	.. (loss)

Table 5. Top 10 terms within <A> tags for "news" category

Rank	Discriminant Efficiency	Term (English translation)
1	0.00705	... (payment)
2	0.00605	...... (inquire)
3	0.00571	.. (merchandise, product)
4	0.00496	... (commercial transaction)
5	0.00483	.. (shopping cart)
6	0.00387	..... (inquire)
7	0.00380	.... (watchlist)
8	0.00380	.... (affiliate link)
9	0.00375	... (set, group of items)
10	0.00358	.. (sundry goods)

Table 6. Top 10 terms within <A> tags for "shopping" category

common terms which appear many times in a document. In Naïve Bayes, frequent terms in a category will have higher weights. If we can collect enough documents for each category that can eliminate the negative effect of low frequencies, we could obtain better results. However, it is impractical to collect such amount of documents to classify over a hundred thousand dimensional vectors. Thus, it might be more practical to consider improving the calculation of term weights.

In the proposed method, we count the pair of a term and a HTML tag as the feature. It is sufficient if we use only one HTML tag for the feature. However, it might be better to consider the nested HTML tags. Besides, in the experiment we use all the terms that appear in the document. Perhaps we can improve the classification by considering stop words and/or restricting the terms by part-of-speech.

Regarding the categories used in the experiments, some categories, such as "commentary" and "internet service", have relatively less characteristic terms compared to other categories. It might be because Web pages in these categories vary in topics, writing styles, and page structures. Other approaches might be necessary for obtaining better results for these

Rank	Discriminant Efficiency	Term (Englishtranslation)
1	0.00116	. (to, into, at, etc.)
2	0.00077	... (to put)
3	0.00077	... (shopping)
4	0.00077	.. ((shopping) cart)
5	0.00038	.. (addition)
6	0.00038	.. ((shopping) cart)
7	0.00038	... (gift)
8	0.00038	. (in, on, at, with, by, etc.)
9	0.00038	.. (to gift)
10	0.00001	.. (display)

Table 7. Top 10 terms within <FORM> and <INPUT> tags for "shopping" category

Rank	Discriminant Efficiency	Term (English translation)
1	0.005066	(<FORM> with no content)
2	0.001514	.. (display)
3	0.000753	(<INPUT> with no content)
4	0.000593	.. (apply)
5	0.000323	submit
6	0.000294	.. (response)
7	0.000294	submitG
8	0.000294	O
9	0.000009	.. (search)
10	0.000006	. (to, into, at, etc.)

Table 8. Top 10 terms within <FORM> and <INPUT> tags for "bbs" category

	SVM	Naïve Bayes
Only term frequencies	61.06%	60.70%
Term freq. and HTML tags	**69.56%**	62.23%

Table 9. Classification accuracies using two classification methods with or without HTML tags

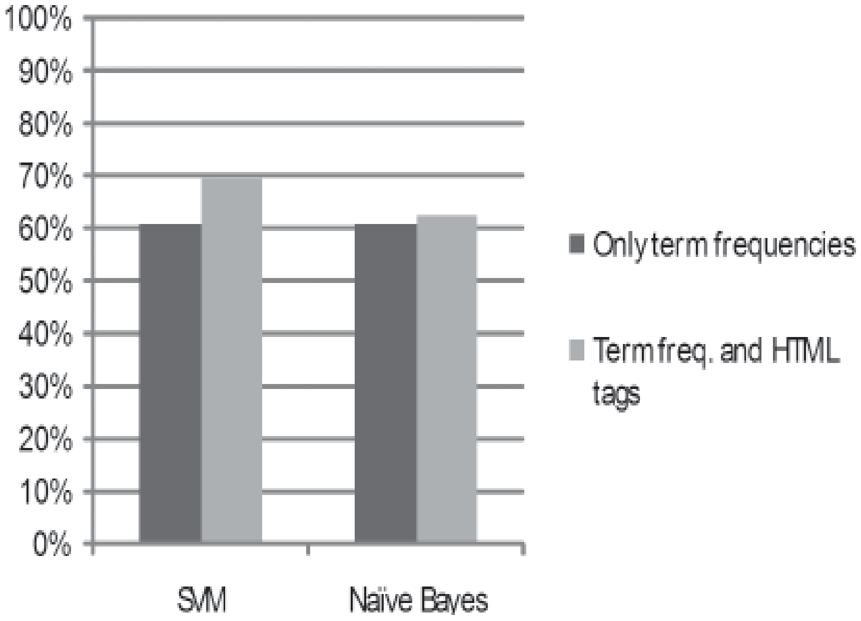

Figure 6. Classification accuracies using two classification methods with or without HTML tags

categories, such as setting a threshold for the likelihood of categories to be classified and considering the documents with lower likelihood to be classified into one of these categories.

7. Conclusion

In this paper, we proposed a method to classify Web documents by genre based on features of terms and HTML tags. In order to improve the accuracy of classification, we calculate discriminant efficiencies of each pair of terms and HTML tags to find out HTML tags which are effective in classification. As the result, we chose to use <TITLE>, <A>, <INPUT>, and <FORM> tags along with the terms themselves. In the experiments of genre classification of Japanese Web documents, our method using term frequencies and HTML tags achieved 8% increase in classification accuracy, in the case of using the SVM classifier.

However, the accuracy is still not far from desirable. In order to achieve practical performance, we are planning to consider the usage of HTML tags in more detail, such as using other HTML tags and nested tags. In addition, it might be useful to consider broader page structure. Web pages of blogs, news sites, BBS, and Wikipedia, etc. usually have some fixed parts surrounding the text body, such as header, footer, side bar, advertisements, etc. In such cases, using only the surrounding parts and disregarding the text body might be effective for genre classification. Other future work includes conducting larger scale experiments, comparable experiments to existing approaches, and extending the proposed method to other languages.

References

[1] Lim, C.S., Lee, K.J., Kim, G.C. (2005). Multiple sets of features for automatic genre classification of web documents. *Information Processing & Management* 41(5) 1263-1276.

[2] Vidulin, V., Lustrek, M., Gams, M. (2007). Training the Genre Classifier for Automatic Classification of Web Pages. *In: Proc. of the 29th International Conference on Information Technology Interfaces (ITI 2007)*, p. 93-98.

[3] Chaker, J., Habib, O. (2007). Genre Categorization of Web Pages. *In:* Proc. of Seventh IEEE International Conference on Data Mining (ICDM 2007) Workshops, p. 455-464.

[4] Levering, R., Cutler, M., Yu., L. (2008). Using Visual Features for Fine-Grained Genre Classification of Web Pages. *In:* Proc. of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), p. 131.

[5] Dong, L., Watters, C., Duffy, J., Shepherd, M. (2008). An Examination of Genre Attributes for Web Page Classification. *In:* Proc. of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), p. 133.

[6] Ferizis, G., Bailey, P. (2006). Towards practical genre classification of web documents. *In:* Proc. of the 15th international conference on World Wide Web, p. 1013-1014.

[7] Santini, M. (2006). Some issues in Automatic Genre Classification of Web Pages. *JADT 06 - Actes des 8 Journées internationales d'analyse statistiques des donnés textuell.*