THE EFFECTS OF DIFFERENT FEATURE SETS ON THE WEB PAGE CATEGORIZATION PROBLEM USING THE ITERATIVE CROSS-TRAINING ALGORITHM

Nuanwan Soonthornphisaj and Boonserm Kijsirikul
Machine Intelligence & Knowledge Discovery Laboratory
Department of Computer Engineering
Chulalongkorn University,
Phathumwan, Bangkok, 10330, Thailand.
Email : nuanwan@mind.cp.eng.chula.ac.th, boonserm@mind.cp.eng.chula.ac.th

Key words: Web page categorization, Iterative Cross-Training, Feature sets

Abstract: The paper presents the effects of different feature sets on the Web page categorization problem. These features are words appearing in the content of a Web page, words appearing on the hyperlinks, which link to the page and words appearing on every headings in the page. The experiments are conducted using a new algorithm called the Iterative Cross-Training algorithm (ICT) which was successfully applied to Thai Web page identification. The main concept of ICT is to iteratively train two sub-classifiers by using unlabeled examples in crossing manner. We compare ICT against supervised naïve Bayes classifier and Co-Training classifier. The experimental results show that ICT obtains the highest performance and the heading feature is considerably succeed in helping classifiers to build the correct model used in the Web page categorization task.

1. INTRODUCTION

Nowadays, there is a massive increase of Web pages in the Internet. An ideal search engine should have the most updated information of all Web pages to provide the best search result for the user. Therefore, it should have an effective Web robot which crawls the Web and automatically classifies Web pages into categories, since Web page classification task is a tedious job and time consuming process if it is done by human. Thus, we want it to be automatic with a reliable classification result.

The problem of text classification has been explored by many researchers with variety of learning algorithms (Cohen & Singer, 1999; Jochim, 1998) When we give a sufficient set of labeled training examples, supervised learning is the most effective method for the classification. However, the construction of hand-labeled data must be done by a human and thus this is a painfully time-consuming process. Though it is costly to construct hand-labeled data, in some domains it is easy to obtain unlabeled ones, such as data in the Internet. Therefore, we propose a new learning algorithm called incremental iterative cross-training (incremental-ICT) in order to utilize the available unlabeled data.

Our incremental-ICT is based on the ICT algorithm that has been successfully applied for identifying Thai Web pages (Kijsirikul et al., 2000). ICT employs two sub-classifiers to iteratively train each other by using unlabeled examples in crossing manner. ICT is based on the assumption that one of the sub-classifiers has some knowledge about the domain. However, this assumption is violated on some domains where we cannot give domain knowledge to the classifier. In such a problem, ICT does not perform well. In this paper, we propose a new algorithm, called incremental-ICT, which requires no such assumption.

To evaluate the robustness of our algorithm, we apply it to a more difficult problem than Thai Web page identification. The problem we are interested in

is the classification of Web pages into course or non-course pages. Since the concept of ICT needs two classifiers, we build each classifier based on different feature sets using naïve Bayes classifiers.

We run experiments to evaluate the effectiveness of our method and to see the contribution of each feature set. In the experiments, we compare our method with the *Co-Training* algorithm (Blum & Mitchell, 1998) and a supervised learning algorithm which uses a naïve Bayes classifier as a classification mechanism. The results show that incremental-ICT gives better performance than the other classifiers.

The paper is organized as follows. Section 2 presents feature sets used in the experiments. Section 3 describes our learning algorithm, and gives the details of a naïve Bayes classifier. Section 4 and 5 describes other learning methods used in our comparison. Section 6 describes the experimental results. Finally, Section 7 concludes our work.

2. FEATURE SETS

For the classification problem, the classifier's performance usually depends on the classification mechanism with the support of feature sets. The appropriate feature sets will help the classifier to enhance its classification correctness. Therefore we try to investigate the possible feature sets to see their contribution on the precision and recall of the classifier. Feature sets that we study are as follows.

2.1 Hyperlink

The Web page in the Internet is a special document. It has a unique characteristic which makes it different from other plain text documents. Most Web pages have hyperlinks that act like a pointer pointing to other pages and also have links from other pages pointing to them. In our case, we use the hyperlink which link to the page to be the first feature set.

2.2 Content

The content of a Web page provides information to the user in detail. We extract all words in the content to be the second feature set.

2.3 Heading

The heading phrase normally represents the main idea of the following content. We use this opportunity to extract all headings in the page in the hope that they could represent the main concept of a Web page.

3. INCREMENTAL ITERATIVE CROSS-TRAINING

The architecture of our learning algorithm consists of two naïve Bayes classifiers, each of which learns from different features of a Web page. For the ease of explanation, we will use the concrete example of feature sets which are words on hyperlinks linking to the page (hyperlink-based) and words on the page (content-based). Starting with a small number of labeled data, each classifier estimates its parameters and uses the learned parameters to classify unlabeled data for the other as shown in Figure 1. The classification for unlabeled data is done in incremental way, i.e., the algorithm incrementally labels a small number of data. The training data is duplicated into two sets: *TrainingData1* for training the hyperlink-based classifier and *TrainingData2* for training the content-based one. The concept of our algorithm is that if we could obtain reliable statistical information from the first classifier, it should be useful in classifying training data for the second classifier. After receiving training from each other, the parameters of the classifiers should be more reliable every iteration.

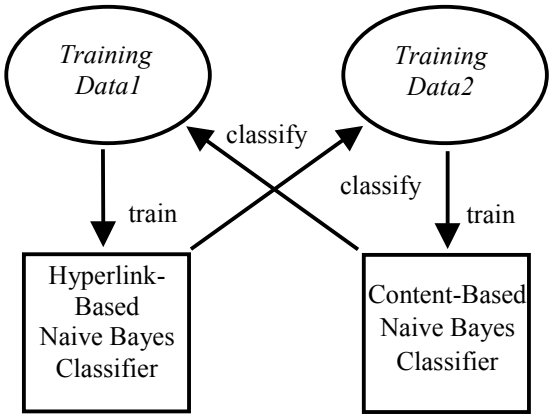

Figure 1: The architecture of iterative cross-training.

The training algorithm of incremental-ICT is shown in Table 1. As shown in the table, the training process starts with the parameter estimation of both classifiers, i.e. hyperlink-based and content-based, using initial labeled data. For each round of iteration, the content-based classifier with the current parameter, θ_c , will classify training data into positive and negative examples. Then it will ask for the confirmation from the hyperlink-based classifier that considers another view of each example to make decision about which class the example should be. If both classifiers agree with the same classification result, the most confident p positive and n negative examples will be labeled.

Table 1: The Incremental-ICT algorithm

Given:

- Two training sets: *TrainingData1* of hyperlink-based data and *TrainingData2* of content-based data (*TrainingData1* and *TrainingData2* both contain *U* labeled examples).

– Use labeled data in *TrainingData1* to estimate the parameter set θ_h of the hyperlink-based classifier.

– Use labeled data in *TrainingData2* to estimate the parameter set θ_c of the content-based classifier.

– Loop until all data are labeled.

- Use the content-based classifier with current θ_c to classify *TrainingData1* into positive and negative examples.
  - Check consistency of the classification with the hyperlink-based classifier. Label the class for the most confident *p* positive examples and most confident *n* negative examples.
- Train the hyperlink-based classifier by the labeled examples in *TrainingData1* to estimate the parameter set θ_h of the classifier.
- Use the hyperlink-based classifier with current θ_h to classify *TrainingData2* into positive and negative examples.
  - Check consistency of the classification with the content-based classifier. Label the class for the most confident *p* positive examples and most confident *n* negative examples.
- Train the content-based classifier by the labeled examples in *TrainingData1* to estimate the parameter set θ_c of the classifier.

The hyperlink-based classifier is then trained by the labeled examples in *TrainingData1* to estimate the parameter set θ_h. With this current θ_h, the hyperlink-based classifier will classify *TrainingData2* into positive and negative examples. Then the consistency checking process is performed again to ask for the agreement from the content-based classifier. The most confident *p* positive and *n* negative examples will be labeled. The content-based classifier starts again with parameter estimation by using labeled examples in *TrainingData1*. These processes will be repeatedly done until all data are labeled.

The classification mechanisms of these two classifiers ar the same which use the naïve Bayes algorithm. This algorithm is a well-known approach and is considered to be one of the most effective way for text classification (Mitchell, 1997) The algorithm employs *bag-of-words* to represent the document. The method is described below.

Given a set of class labels L = $\{l_1, l_2, \ldots, l_m\}$ and a document d of n words $(w_1, w_2, .., w_n)$, the most likely class label l^* estimated by naïve Bayes is the one that maximizes $Pr(l_j|w_1, \ldots w_n)$:

$$l^* = \underset{l_j}{argmax}\ Pr(l_j|w_1, \ldots, w_n) \quad (1)$$

$$= \underset{l_j}{argmax}\ \frac{Pr(l_j)Pr(w_1, \ldots, w_n|l_j)}{Pr(w_1, \ldots, w_n)} \quad (2)$$

$$= \underset{l_j}{argmax}\ Pr(l_j)Pr(w_1, \ldots, w_n|l_j) \quad (3)$$

For our data set, L is the set of positive and negative class labels which are course homepage and non-course homepage, respectively. $Pr(w_1, \ldots, w_n)$ in equation 2 can be ignored, as we are interested in finding the most likely class label. As there are usually an extremely large number of possible values for $d = (w_1, w_2, \ldots, w_n)$, calculating the term $Pr(w_1, \ldots, w_n|l_j)$ requires a huge number of examples to obtain reliable estimation. Therefore, to reduce the number of required examples and improve reliability of the estimation, assumptions of naïve Bayes are made. These assumptions are (1) the conditional independent assumption, i.e. the presence of each word is conditionally independent of all other words in the document given the class label, and (2) an assumption that the position of a word is unimportant, e.g. encountering the word "subject" at the beginning of a document is the same as encountering it at the end (Mitchell, 1997). Equation 3 can be rewritten as:

$$l^* = \underset{l_j}{argmax}\ Pr(l_j) \prod_{i=1}^{n} Pr(w_i \mid l_j, w_1, \ldots, w_{i-1}) \quad (4)$$

$$= \underset{l_j}{argmax}\ Pr(l_j) \prod_{i=1}^{n} Pr(w_i \mid l_j) \quad (5)$$

The probabilities $Pr(l_j)$ and $Pr(w_i|l_j)$ are used as the parameter sets θ_h and θ_c, and are estimated from the training data. The prior probability $Pr(l_j)$ is estimated as the ratio between the number of examples belonging to the class l_j, and the number of all examples. The conditional probability $Pr(w_i|l_j)$, of seeing word w_i given class label l_j, is estimated by the following equation:

$$Pr(w_i|l_j) = \frac{1 + N(w_i, l_j)}{T + N(l_j)} \quad (6)$$

Where $N(w_i, l_j)$ is the number of times word w_i appears in the training examples from class label l_j, N

(l_j) is the total number of unique word in the training set. T is the number of class. Equation 6 employs Laplace smoothing (add one to all of word counts), to avoid assigning probability values of zero to words that do not occur in the training examples for a particular class.

To evaluate our method, we will compare it with the other two techniques that are the Co-Training and the supervised naïve Bayes classifiers. These classifiers are described in the following sections.

4. CO-TRAINING CLASSIFIER

The Co-Training algorithm explicitly uses the split of the features when learning from labeled and unlabeled data. Its approach is to build the naïve Bayes classifier for each of the distinct feature sets. Each classifier is initialized using a few labeled documents. Then every round of Co-Training, each classifier chooses the most confident p positive and n negative labeled examples to add to the labeled set of documents. The documents selected are those that have the highest posterior class probability, $Pr(l_j|d)$. Then, each classifier rebuilds from the augmented labeled set and the process repeats (Blum & Mitchell, 1998).

Table *2*: The Co-Training algorithm

Given:
A set *LE* of labeled training examples
A set *UE* of unlabeled examples
Create a pool *UE'* of examples by choosing u examples at random from *UE*.
Loop while there exist documents without class labels:

- Use *LE* to estimate θ_h of the hyperlink-based classifier using the hyperlink portion of each document.
- Use *LE* to estimate θ_c of the content-based classifier using the page portion of each document.
- Allow the hyperlink-based classifier with current θ_h to label p positive and n negative examples from *UE'*.
- Allow the content-based classifier with current θ_c to label p positive and n negative examples from *UE'*.
- Add these self-labeled examples to *LE*.
- Randomly choose $2p+2n$ examples from *UE* to replenish *UE'*.

5. SUPERVISED NAÏVE BAYES CLASSIFIER

The basic concept of *supervised learning* for building a classifier is that it requires a set of examples with predefined classes. The classifier is then try to find some common properties of the different classes in order to make correct classification for unseen data. Thus, this kind of classifiers need a large number of labeled examples to correctly model the characteristic of the class during learning process. Labeling must be done by human to train the classifier accurately. In our experiment, we employ the naïve Bayes classifier as a supervised learning algorithm. The algorithm of the naïve Bayes is the same as one described in Section 3, except that it is trained by hand-labeled data.

6. EXPERIMENTAL RESULTS

In order to test the robustness of the incremental-ICT algorithm and to investigate the effectiveness of feature sets , we set up experiments on the problem of course/non-course Web page classification, and compare the performance of incremental-ICT to the other classifiers, i.e., the Co-Training algorithm and the supervised naïve Bayes classifier.

6.1 Data Set

The data for our experiments is obtained via ftp from Carnegie Mellon University (The World Wide Knowledge Base Project). It consists of 1,051 Web pages collected from Computer Science department Web sites at four universities: Cornell, University of Washington, University of Wisconsin, and University of Texas. These Web pages have been hand-labeled into two categories. We consider the category "course home page" as the positive class and the other as the negative class. In this dataset, 22% of the Web pages were course homepages and the rest were non-course homepages.

In this data set, the two Web categories are actually closely related which make the classification more difficult. A course home page gives information about the subject such as the course outline, the class schedule, reference books. A non-course homepage is an instructor homepage or department Web page.

6.2 Experimental Setting

We have 230 course Web pages and 821 non-course Web pages. Each sample is filtered to remove words which give no significance in predicting to the class of the document. Words to be eliminated are auxiliary verbs, prepositions, pronouns, possessive

pronouns, phone numbers, digit sequences, dates and special characters. The training set contains 172 course Web pages and 616 non-course Web pages.

Three positive examples and nine negative examples were randomly selected from the training dataset to be initial labeled data. Therefore, each data set contains 12 initial labeled examples, 776 training examples and 263 testing examples. We then used 3-fold cross-validation (Mitchell, 1997) for averaging the results. Three positive and nine negative samples are used as the initial labeled data for the incremental-ICT and the Co-Training algorithms. The parameters *p* and *n* in Table 1 and Table 2 is set to 1 and 3, respectively.

6.3 The Results

Standard precision (P), recall (R), accuracy (A) and F1-measure (F1) are used to evaluate the performance of the classifiers. These are defined as follows.

$$P = \frac{\text{no. of correctly predicted positive examples}}{\text{no. of predicted positive examples}}$$

$$R = \frac{\text{no. of correctly predicted positive examples}}{\text{no. of all positive examples}}$$

$$A = \frac{\text{no.of correctly predicted examples}}{\text{no. of all examples}}$$

$$F1 = \frac{2PR}{P+R}$$

6.4 Experiment using content and hyperlink features

For the first experiment, we use words appearing in the content of a Web page as the feature for the first classifier. The second classifier uses words appearing on the hyperlink as a feature set. The results are shown in Table 3.

Table 3: Performance of content-based and hyperlink-based classifiers using 3-fold cross-validation: P = Precision, R = Recall, A = Accuracy, F1 = F1-measure.

Classifier	P(%)	R(%)	A (%)	F1
I-ICT (content)	94.04	80.46	94.55	86.72
S-Bayes (hyperlink)	85.34	63.22	89.48	72.61
I-ICT (hyperlink)	67.54	72.41	85.17	69.89
Co-Training (content)	81.52	54.08	87.32	65.03
S-Bayes (content)	99.05	42.20	87.25	58.97
Co-Training (hyperlink)	75.92	44.83	84.28	56.37

In Table 3, I-ICT (content) and I-ICT (hyperlink) stand for the content-based and hyperlink-based naïve Bayes classifiers of the incremental-ICT, respectively. Co-Training (content) and Co-Training (hyperlink) are content-based and hyperlink-based naïve Bayes classifiers of the Co-Training algorithm, respectively. S-Bayes (content) and S-Bayes (hyperlink) are supervised naïve-Bayes classifiers, which classify Web pages based on words in Web pages and words in hyperlinks, respectively.

As shown in the table, I-ICT (content) gives the best performance followed by S-Bayes (hyperlink), I-ICT (hyperlink), Co-Training (content), S-Bayes (content) and Co-Training (content), respectively. The reason that I-ICT (content) gives better performance compared to S-Bayes is because I-ICT (content) cooperates with I-ICT (hyperlink) while S-Bays uses single classifier. The performance of I-ICT (hyperlink) is not as good as that of I-ICT (content). This is because hyperlinks contain fewer words and thus are less capable of building the accurate classifier. The training technique of I-ICT is also an effective way, as its performance is better than that of Co-Training, which uses a different training technique.

6.5 Experiment using content and heading features

In order to see the impact of the heading feature on the categorization problem, we did experiments using heading-based classifier with various learning algorithms.

As shown in Table 4, I-ICT (content) and I-ICT (heading) stand for the content-based and heading-based naïve Bayes classifiers of the incremental ICT algorithm. S-Bayes (heading) and S-Bayes (content) are supervised naïve Bayes classifiers based on heading and content features, respectively.

Co-Training(heading) and Co-Training (content) are the heading-based and content-based naïve Bayes classifiers of the Co-Training algorithm.

Table 4: Performance of heading-based and content-based classifiers using 3-fold cross-validation: P = Precision, R = Recall, A = Accuracy, F1 = F1-measure.

Classifier	P(%)	R(%)	A (%)	F1
I-ICT (content)	98.12	78.66	95.05	87.32
S-Bayes (heading)	96.51	77.58	94.43	86.02
I-ICT (heading)	80.72	89.65	92.65	84.95
Co-Training (heading)	79.71	74.71	90.24	77.13
Co-Training (content)	82.49	43.68	85.93	57.11
S-Bayes(content)	99.05	42.20	87.25	58.97

The best performance belongs to the content-based naive Bayes classifier of I-ICT followed by the supervised naïve Bayes classifier based on the heading feature, the heading-based of I-ICT, the heading-based of co-training, the content-based of co-training and the content-based of the supervised naïve Bayes classifier.

7. DISCUSSION AND CONCLUSION

In this paper, we have demonstrated the concept of the I-ICT algorithm and investigate the impacts of feature sets to the classification correctness. From the experimental results, the performance of S-Bayes (heading) is much higher than that of S-Bayes (hyperlink). It means that the heading feature has more potential than the hyperlink feature in helping the classifier to build the correct model used in categorization task. This is because the detail of the Web page is usually organized into sub-sections with the headings, which represent the main idea of the following content. Thus the structure of all headings in a page should give some common properties that is useful to identify its category. In the contrary, the words in the hyperlink, which links to the page could not provide enough information to identify the class of the page. This is not surprising because normally the hyperlink phase contains just few words referring to the page. For the worst case, the hyperlink might contain only a proper noun that is not sufficient in classifying that referring page.

According to the F1-measure, we obtained only 58.97% accuracy for S-Bayes using the content feature. It implies that the content feature alone could not help much in Web page classification because the two Web categories actually have high relevance in detail. Therefore the naïve Bayes classifier could not find the exact model for each category using all words appearing in the page.

Considering all classification mechanisms, we found that our I-ICT algorithm provides the highest correctness in both experiments. This is because I-ICT combines two classifiers based on different feature sets and these two classifiers cooperate with each other during the training process. The I-ICT algorithm has been proved to be robust under new assumption that each example can be viewed in two different views using different feature sets. With the consistency checking process, which is used to compensate the lack of domain's knowledge of the classifiers, our algorithm outperformed the supervised naïve Bayes algorithm, and Co-Training algorithm.

ACKNOWLEDGEMENT

This paper is supported by the Thailand Research Fund and National Electronics and Computer Technology Center (NECTEC) under the project number NT-B-06-4F-13-311.

REFERENCES

Kijsirikul, B., Sasipongpairoege, P., Soonthornphisaj, N. and Meknavin, S., 2000, 'Supervised and Unsupervised Learning Algorithms for Thai Web Pages Identification', *Proceeding of the Pacific Rim International Conference on Artificial Intelligence* (PRICAI-2000), 690-700.

Blum, A. and Mitchell, T., 1998, 'Combining Labeled and Unlabeled Data with Co-Training', *Proceeding of the Eleventh Annual Conference on Computational Learning Theory* .

Cohen, W. and Singer, Y., 1999, 'Context-sensitive learning methods for text categorization', *ACM Transactions on Information Systems*, 17(2): 141-173.

Joachims, T., 1998, 'Text categorization with support vector machines: Learning with many relevant feature', *Proceedings Tenth European Conference on Machine Learning*, Springer Verlag.

Nigam, K., McCallum, A., Thrun, S. and Mitchell, T., 2000, 'Text classification from labeled and unlabeled documents using EM', *Machine Learning*, 39(2/3): 103-134.

Apte, C., and Damerau, F., 1994 'Automated learning of decision rules for text categorization', *ACM TOES*, 12(2): 233-251.

Mitchell, T., 1997, *Machine Learning*, McGraw-Hill. New York, 180-184.

The World Wide Knowledge Base (web-kb) project, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/course-cotrain-data.tar.gz, Carnegie Mellon University, U.S.A.