Web Document Classification based on Tagged-Region Progressive Analysis

Li-Chun Sung
Computers and Networking (CAN) Laboratory
Department of CS&IE
Tamkang University, R.O.C.
g7190030@tkgis.tku.edu.tw

Meng-Chang Chen
Advanced Internet Protocols and Services Group
Institute of Information Science
Academia Sinica, R.O.C.
mcc@iis.sinica.edu.tw

Chin-Hwa Kuo
Computers and Networking (CAN) Laboratory
Department of CS&IE
Tamkang University, R.O.C.
chkuo@mail.tku.edu.tw

Abstract*-In this paper, we propose an intelligent web document classification method, called TAgged-Region Progressive Analysis (TARPA). Instead of parsing the whole content of the web page while classifying a web document, TARPA parses the document into finer structured Tagged-Regions and extracts fewer and the most important regions to analyze and classify. If the few important tagged regions are not sufficient to allow TARPA to classify the document, other important regions and linked pages can be used for analysis progressively to enhance the classification performance. TARPA possesses two stages: learning stage and classification stage. The learning stage discriminates the importance of tag-pairs, and the classification stage follows the importance order of tag-pairs to analyze the document. As a result, TARPA can classify a web document using few contents while with higher classification rate and shorter processing time. Experiments show that 91% of the testing web documents can be correctly classified by only feeding the TARPA classifier with 40% to 50% of the document contents.*

Keywords: *W*eb categorization, Progressive Analysis

1. Introduction

Web documents are written in *Hyper Text Markup Language* (HTML), mostly generated by document composers. In HTML, tags or pairs of tags (tag-pairs) are used to tell the computer how to treat individual expressions and how to construct the document. Through the use of HTML tags, a document can be constructed and made to appear exactly as the original shown in document composer. Almost all tags have some specific uses in the whole HTML document. Expressions in the HTML document are enclosed in tag-pairs to have or represent certain properties. For example, terms that are enclosed in the TITLE tag-pair often display the title of the web document; terms that are enclosed in the *Header Tags* H1~H6 are often used to mark the importance of the article and to lay stress on the key points in the web document. Consequently, terms that are in specific text areas usually hold more significance in helping readers grasp the true meanings of web documents. However, many web documents contain too much information, including advertisements, links to other un-related web pages, other titbits (such as "joke of the day" reflecting the author's personality), that hinders an automatic classifier in correctly classifying web documents. Hence, understanding the meaning and importance of tags and their combinations is essential in automatic classification.

1.1. Related works

Besides traditional text analysis methods, classification systems have, in recent years, begun to use the features of web documents to improve the precision of classification. Most of the related researches focus on one of the two areas: (1) HTML language structure, and (2) linking architecture. The researches that focus on the former, analyze documents according to the contained HTML tags. They often need to make some assumptions on the tags contained in the documents. For example, the *LDA* algorithm [1] assumes that the contained tags of a web document are deployed based on a predefined style; *Voting* scheme [2] and *Webfoot* scheme [3] assume that these contained tags are ordered based on a predefined rank list; and *ACIRD* [4] assumes each tag is pre-assigned with a weight. Therefore, these classification systems can not work when the web document style changes or the next generation of the markup language standard, like XML language, appears. In addition, the schemes are not effective when the web document itself does not contain enough information to be classified.

The other approaches use the linking pages of web documents to help document analyses. These systems [5-8] first download linking pages within the specific range according to the linking graph of the web documents, and they merge the classification results of these downloaded linking pages to figure out the eventual category. However, they may

be bound by large networking overhead, especially when it requires heavy downloading of linked pages.

To achieve the goals of efficiency and high precision, in this paper we propose a novel progressive analysis scheme for web document classification: TAgged-Region Progressive Analysis scheme, which is abbreviated as TARPA. The paper is outlined as follows: In section 2, we present the architecture of the TARPA. In section 3, further we describe the operation detail of the proposed classifier. In section 4, we use experiment results to show that the TARPA classifier is better than others. We conclude our present work and point out future research direction in section 5.

2. The TARPA system architecture

In order to alleviate the problem of the traditional classifier, we propose the TARPA classifier with progressive content analysis. The system architecture and the functions are shown in Fig. 1. The classifier consists of major components: (1) *Progressive Analyzing* unit, (2) *Categorization Reference Database*, (3) *Tag-based Regionalizing* unit, (4) *Feature Analyzing* unit, and (5) *Similarity Calculating* unit.

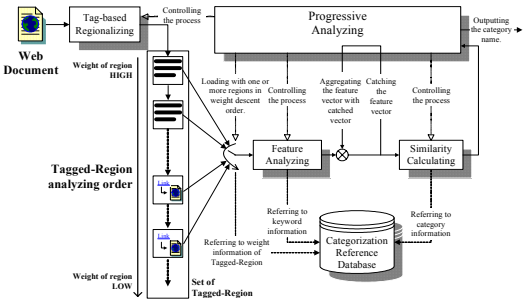


Figure 1. The architecture of the TARPA classifier

Among these five components, the *Progressive Analyzing* unit is the control center of the whole classifier. It coordinates the other four units to perform the content analysis and decides which category the document belongs to. During web document analyzing, the main controller controls the *Tag-based Regionalizing* unit to partition the whole document into smaller structure-based regions, named *Tagged-Regions*. Then the *Feature Analyzing* unit and *Similarity calculating* unit analyze these regions possessively in descending order of tagged-region weights until the similarity measurement between the document and one or more categories is higher than the pre-defined threshold, or fail to classify the document. Because meaningful paragraphs are representative of the document, we arrange to have them analyzed as a high priority. If these meaningful paragraphs contain enough information, we can find the correct category early, and thus reduce the time of analyzing. However, if the web document itself does not contain enough meaningful information, we further regard the contents of linked pages, following the order of their weights, as a single *Tagged-Region* to assist the similarity calculation.

To further explain our proposed classification strategy, in the following sections, we describe the core *Tagged-Region* concept and the Categorization *Reference Database* that maintains the required parameters for classification.

2.1. The concept of Tagged-Regions

The concept of the proposed classifier is based on analyzing the text areas delimited by tag-pairs. In order to explain the theory clearly, we identify the set of words enclosed in a tag-pair as *Tagged-Region*, $\Re$. We also define the following expression to describe this notion:

Let $\boldsymbol{p}$ be a web document,

Φ be the set of tag-pairs defined in HTML 4.0,

Φ_p be the set of tag-pairs that appear in $\boldsymbol{p}$,

$$\Re_{p,i}^{<\beta>} \quad \beta \in \Phi_p \ , \ \Phi_p \subseteq \Phi \quad \text{and} \quad i = 1, 2, \ldots, \left|\Re_p^{<\beta>}\right|, \quad (1)$$

where β is the tag name of the tag-pair that form the enclosure, $<\beta>$ indicates the type of $\Re$, $\left|\Re_p^{<\beta>}\right|$ represents the number of regions of type $<\beta>$ in $\boldsymbol{p}$, and i indicates the appearance order of this kind of regions in $\boldsymbol{p}$.

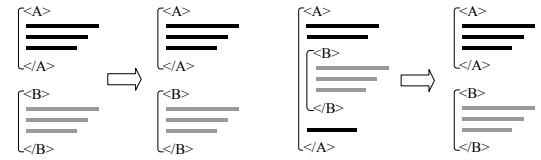
Besides the physical characteristics of $\Re$, we also have to consider the influence of the space layout of $\Re$s. In general, as shown in Fig. 2, the space layouts of web documents usually can be divided into two categories: (a) parallel, and (b) nested. In the parallel space layout, $\Re$s are independent, and can be dealt with individually. In the nested space layout, on the contrary, an $\Re$ may contain other $\Re$s, and so on. In this paper, the regions that contain other regions are referred to as outer *Tagged-Regions*, and the contained regions are referred to as inner *Tagged-Regions*. Portions of the content of web document could be shared by one of the inner $\Re$s and several of the outer $\Re$s simultaneously. These regions intuitively own their appropriate influence according to the features of their individual enclosing tag-pairs in this overlap content. We consider these overlap content principally based on the features of the inner $\Re$. In the case of a mixed parallel space layout and nested space layout, called mixed space layout, it makes use of the rule of the parallel and nested space layouts simultaneously, (see Fig. 2).

With regard to the nested space layout, we observe an interesting phenomenon. The significance of the inner $\Re$s will be accordingly enhanced when they match up with some appropriate outer $\Re$s. Consequently, the inner $\Re$s of a region type in the nested and mixed space layouts could have different importance with different outer $\Re$s. If we consider the im-

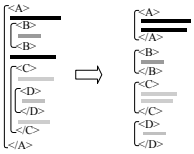
portance of these inner ℜs are identical, we will lose a lot of information that facilitates text classification. Therefore, we extend the original definition of ℜs in order to distinguish the inner regions that have a different importance. An expression of the *extended Tagged-Region* type is defined as follows:

$$< \alpha < \beta >> \qquad \alpha, \beta \in \Phi_p \ , \ \Phi_p \subseteq \Phi \tag{2}$$

where β expresses the tag name of the inner region and is the base type; and α is the tag name of the outer region and is the extended type.



(a) Parallel space layout (b) Nested space layout



(c) Mixed space layout

Figure 2. Space layouts of the *Tagged-Region*s

The above expression represents the type of segment in the web document by means of the two tag names. The reasons that we do not consider presently more enclosing tag-pairs of a nested tagged region are; (1) we can not collect enough effective samples to train the classifier, (2) many of the fastidious types of ℜs may have the same importance, and (3) the computing overheads may increase substantially. By combining Eq. (1) and Eq. (2), the expression for the collection of terms of the extended *Tagged-Region* type is defined by:

$$\Re_{p,i}^{<\alpha<\beta>>} \qquad i = 1, 2, \ldots, \left|\Re_p^{<\alpha<\beta>>}\right|, \tag{3}$$

where $\left|\Re_p^{<\alpha<\beta>>}\right|$ expresses the number of regions of type $< \alpha < \beta >>$ in $\boldsymbol{p}$.

2.2. Categorization Reference database

In general, to be able to analyze and classify a web document, a classifier must be provided with related reference information. In our TARPA scheme, we maintain these required parameters in the *Categorization Reference* database.

Among these reference data, four common fundamental types of data are defined as follows:

Let C be the set of web document categories,

$|C|$ be the size of the set C,

ω_{ref} be the set of collected keywords,

$|\omega_{ref}|$ be the size of the set ω_{ref},

Furthermore, the database also maintains some classification-related information. In the proposed classifier, we adopt the well-known document representation scheme, named *Vector Space Model* (VSM) [9], to be used in calculating the similarities between feature vectors. The delegate feature vector of each category consists of $|\omega_{ref}|$ elements, and each element indicates the importance of the category-related importance of a specific keyword collected in ω_{ref}. The degree of a keyword's importance is calculated based on the popular *TF-IDF* concept. We use the average *Term Frequency*(TF) of the specific keyword in the category and the unique *Inverse Document Frequency*(IDF) of this keyword to produce the element value. These classification-related parameters are defined as follows:

Let $TF_{C_j}(w)$ be the average normalized *TF* value of word w in category C_j, $IDF(w)$ be the *IDF* value of keyword w, $W_{C_j}(w)$ be the weight of keyword w in C_j that $W_{C_j}(w_\gamma) = TF_{C_j}(w_\gamma) \times IDF(w_\gamma)$, $\vec{i}_r$ be the γth row vector of $|\omega_{ref}| \times |\omega_{ref}|$ unit vector, $\vec{f}_{C_j}$ be the delegate feature vector of C_j that

$$\vec{f}_{C_j} = \sum_{\gamma=1}^{|\omega_{ref}|} \vec{i}_\gamma \times W_{C_j}(w_\gamma) \ \ C_j \in C, \ w_\gamma \in \omega_{ref}.$$

In addition, the database contains information about the weight list of ℜs. This list is used to decide the analyzing order of ℜs in the web document. We define the weight parameter as follows:

$W(\Re^{<\alpha<\beta>>})$ is the weight of $\Re^{<\alpha<\beta>>}$.

3. Web document progressive analyzing

In the section, we explain the operating details of the progressive analyzing concept, which is performed by *Progressive Analyzing* unit.

3.1. Progressive Analyzing

Instead of analyzing the entire content of a web document, as described in section 2, we develop a novel analyzing scheme for web documents. In the proposed scheme, we only analyze one or a few ℜs with higher weights at first. If these anticipated- important regions are real meaningful ones, we can extract more conspicuous features from them than from the entire document because they contain a small amount of meaningless information only. Furthermore, the computing time can be reduced to a large extent.

But, if the extracted features are not enough for deciding the eventual category, the *Progressive Analyzing* unit will load other ℜs in descending order of

the weights of their region types and deliver them for analysis. The new extracted features will be aggregated with the existent features and then the aggregation will be delivered to calculate the similarities with each category. The aggregating operation is expressed as following:

Let $\vec{f}_p^{old}$ be the existent feature vector, $|\vec{f}_p^{old}|$ be the total number of words contained in analyzed $\Re$s, $\Re_{p,i}^{<\alpha<\beta>>}$ be the region that will be analyzed next, $\vec{f}_{p,i}^{<\alpha<\beta>>}$ be the feature vector of $\Re_{p,i}^{<\alpha<\beta>>}$,

$$\vec{f}_p = \vec{f}_p^{old} \times \frac{|\vec{f}_p^{old}|}{|\vec{f}_p^{old}| + |\widetilde{\Re}_{p,i}^{<\alpha<\beta>>}|} + \vec{f}_{p,i}^{<\alpha<\beta>>} \times \frac{|\widetilde{\Re}_{p,i}^{<\alpha<\beta>>}|}{|\vec{f}_p^{old}| + |\widetilde{\Re}_{p,i}^{<\alpha<\beta>>}|}.$$

With regard to the endpoint of the analyzing process, we set a threshold value ($T_{Threshold}$) to confirm the eventual category C_j if the highest similarity can satisfy the following condition:

$$\cos(\vec{f}_{C_j}, \vec{f}_p) \quad \leq \quad T_{Threshold}.$$

3.2. Weight of *Tagged-Region* Type

The weight of a region type is positive proportioned to the amount of meaningful information that it provides. In general, because a region that have a lot of meaningful information is more similar to it's category, we adopt the similarity between the region and it's category to represent this kind of amount. In addition, because the same type of $\Re$s can appear multiple times in the same document, different documents, and different categories, we calculate the weight of any type of $\Re$ by averaging the average similarity of the same type of $\Re$s of each category. And these sources are created through averaging the average similarity of the same type of $\Re$ of each document in the same category. The definition of the weight of $\Re^{<\alpha<\beta>>}$, $W(\Re^{<\alpha<\beta>>})$, is expressed as follows:

$$W(\Re^{<\alpha<\beta>>}) = \underset{j}{avg}(W_{C_j}(\Re^{<\alpha<\beta>>}))$$

where $W_{C_j}(\Re^{<\alpha<\beta>>}) = \underset{\gamma}{avg}(W_{C_j,P_\gamma}(\Re^{<\alpha<\beta>>}))$, and

$$W_{C_j,P_\gamma}(\Re^{<\alpha<\beta>>}) = \underset{i}{avg}(\cos(\vec{f}_{C_j,P_\gamma,i}^{<\alpha<\beta>>}, \vec{f}_{C_j})).$$

4. Experiments and analyses

To show the capability of the TARPA classifier, we perform experiments and show the results in this section. Besides the efficiency of classification, we also present the training result to verify the assumption made by the *Tagged-Region* concept.

In this paper, we adopt three categories of training and testing web documents, including: (1) Image Compression (labeled C_1), (2) ATM (Asynchronous Transfer Mode) Network (labeled C_2), and (3) Wireless Network (labeled C_3). For each category, we query the search engine *Google* to obtain related web documents. For each category, we select 100 related web documents from the collection set as training data, and 300 web documents of the collection set are treated as testing data.

4.1. Classifier training

Based on the experiment setting, the TARPA system detects 319 types of $\Re$s in total through analyzing the training documents. In these types, there are 81 types without text content and the weights of these types of $\Re$s are set to be zero, such as <BODY<TABLE>>, and <TABLE<TR>>. In Table. 1, we list partial region types with their weights. From the training result, we find that the resulting weights reflect the web document writing convention. The type weight is mainly influenced by the significance of the inner tag-pair even when with the same outer tag-pair. Observing the regions whose outer tags are <BODY> tags in Table. 1, we find that the weight order of $\Re$s: <BODY<H1>> > <BODY<H2>> > … > <BODY<H6>> is consistent with the conventional significance order of tags: <H1> > <H2> > … > <H6>.

In addition, through the partial weight list, we can confirm the assumption that the importance of region types which have the same inner tag is not identical when they have different outer tags. For example, the table shows explicitly that the different outer tags will immediately affect the importance of the inner tag <H1>. For example, <CENTER<H2>>, <TD<H2>> and <BODY<H2> are more important than <A<H1>>; <CENTER<H4>> and <DIV<H4>> are more important than <P<H2>> and <A<H2>>.

Table.1 The partial weight list of region types whose inner tags are *Header Tag*s.

Tagged-Region	Weight	Tagged-Region	Weight
<STRONG<H1>>	0.406773	<A<H1>>	0.116138
<BODY<H1>>	0.341776	<CENTER<H4>>	0.110860
<TD<H1>>	0.291320	<DIV<H4>>	0.105342
<P<H1>>	0.276695	<P<H2>>	0.094463
<CENTER<H1>>	0.259732	<A<H2>>	0.092872
<CENTER<H2>>	0.206377	<BODY<H3>>	0.035960
<TD<H2>>	0.159648	<BODY<H5>>	0.012025
<A<H3>>	0.136158	<BODY<H6>>	0.005458
<BODY<H2>>	0.129756		

4.2. Classifier testing

The TARPA scheme is a novel web document analysis methodology that has better performance in classification for most types of Web documents. We discuss the characteristics of the similarity curves of

different types of Web documents in section 4.2.1. In addition, we present the classification performance in section 4.2.2.

4.2.1. Characteristics of similarity curve

In the following, we present the similarity curves of four types of web documents which possess distinct styles of document space layouts. We show the results of the TARPA against category *j* (labeled "C_j-TARPA") with the traditional classification method which analyzes sequentially documents from the beginning of the document text (labeled "C_j-Sequential").

For each similarity curve type, we present two figures, which include:

1. The similarity curve based on the amount of analyzed ℜs (Fig. (a))
   (In this figure, the x-axis represents the amount of analyzed ℜs; the y-axis represents the similarity between ℜs and the specified category.)
2. The similarity curve based on the amount of analyzed keywords (Fig. (b))
   (In this figure, the y-axis represents the similarity between the keywords and the specified category.)

In both Fig (a) and (b), we present the similarity curves of categories C_1, C_2 and C_3 for comparison.

In general, the ideal similarity curves of the proposed scheme should approximate to Fig. 3. It consists of two zones, named Zone A and Zone B. Since the most meaningful ℜs are usually analyzed first, the accumulated similarity should grow rapidly, as in Zone A. But, while the number of analyzed ℜs increases, noise is also introduced. This phenomenon makes the similarity grow slowly, and even decline gradually, as shown in Zone B. Therefore, the peak similarity should appear at the end of Zone A, which is the most appropriate point to confirm the category of a web document. Furthermore, for noisy web documents, our classifier can use the peak similarity characteristic to promote the classification rate.

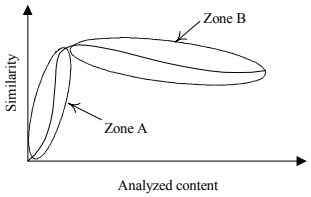
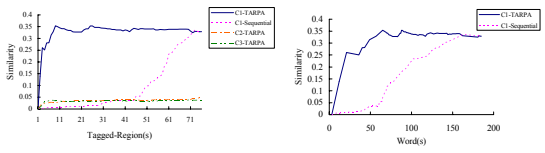
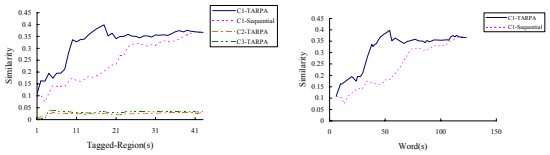


Figure 3. The ideal similarity curve of the progressive classification.



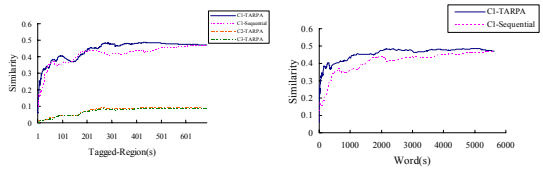
(a) (b)

(1) http://www.zdwebopedia.com/TERM/J/JPEG.html



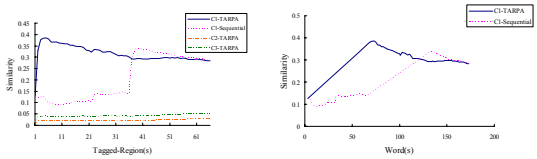
(a) (b)

(2) http://www.fedele.com/website/compress/compress.htm



(a) (b)

(3) http://bmrc.berkeley.edu/research/mpeg/faq/mpeg2-v38/faq_v38.htm



(a) (b)

(4) http://www.bookpool.com/.x/maqsnzkm78/sm/0070633444/index.htm

Figure 4. The classification results.

We show the classification results of the four web documents in Fig. 4. In these results, we adopt Fig. 4(1)~(4) to explain the influences of distinct space layouts on similarity curve:

(1) Keywords mass in a specific portion of the web document

As shown in Fig. 4(1), most of the representative keywords mass at the bottom-right of the document. Based on our proposed scheme, the meaningful contents from the bottom-right portion are directly used for classification, and the peak similarity can be reached rapidly, as shown in Fig. 4(1)(a).

(2) Keywords are distributed over specific portions of the web document

Most keywords are distributed over specific portions of the web document. In this example, keywords are in the hyperlinked strings of the document, as shown in Fig. 4(2). In this situation, the keywords mix with noise. Employing the weights of ℜs, the proposed classifier can intelligently select these meaningful regions to analyze, and the peak similarity can be reached in the early stage, as shown in Fig. 4(2)(a).

(3) Keywords are distributed over the web document.

All of the portions of the document contain some representative keywords, as shown in Fig. 4(3). In this situation, our proposed scheme and the traditional sequential classifier has the same effect, as shown in Fig. 4(3)(a).

(4) Very few keywords reside in a very noisy web document.

This type of web document contains too much unrelated information. From the results, the proposed scheme can produce almost the ideal similarity curve,

like in Fig. 4(4)(a), as it analyzes representative regions first.

In addition, for all four types of web document, as shown in all the (b) sessions of Fig. 4, our proposed classifier only needs 30%~50% of the document content for classification.

4.2.2. Performance of classification

The classification results are arranged and shown in Fig. 5 and Fig. 6. In the two figures, the x-axis represents the amount of analyzed ℜs and content respectively; the y-axis represents the percentage of web documents correctly classified by analyzing a specified amount of ℜs or content. Both of the two figures include three curves: the curve when using the TARPA scheme, the curve when using the sequential scheme, and the curve when using the traditional VSM scheme to perform full-text classification.

In both figures, we show that the partial analysis concept can reduce the classification overhead and promote the classification correction ratio. Our proposed scheme performs extremely well so that 51% of test web documents can be classified by analyzing only 10% of document content, and 91% of test web documents can be classified by analyzing 50% of document content. In addition, for noisy documents, it can reach 95.68% correction, which is higher than the 86.08% of the sequential model and the 79.47% of the traditional VSM model. Summarizing the above comparisons, we prove that our TARPA methodology is superior for web document classification.

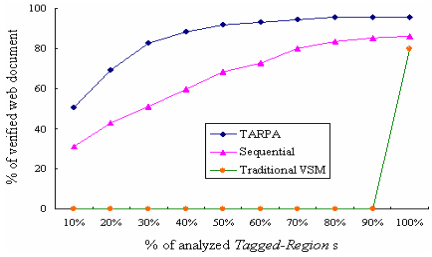


Figure 5. Classification results based on the number of analyzed ℜs.

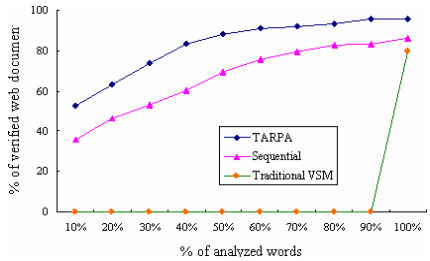


Figure 6. Classification results based on the number of analyzed keywords

5. Conclusion

In this paper, we propose TARPA, a web document classification scheme. Its scheme is designed around the *Tagged-Region* concept which makes it different from the existing classifiers. Based on the principle of "meaningful regions first", the classification of web documents is faster and more accurate than other web document classifiers.

According to the experiment results, we can show that our TARPA scheme can classify various types of web documents rapidly, and only use 50% or less of the document content for classification for the majority of web documents. Furthermore, the TARPA classifier also achieves a higher classification correction rate than other approaches.

The weighting of region types is an evaluation based on the common tag usage. There indeed exist some web documents that disobey the common writing style. We will develop a dynamic weight adjustment scheme to handle the weight variation of tag-pairs in them in the future.

Acknowledgement

This research was partly supported by NSC under projects 92-2213-E-001-002 and 93-2524-S-001-001.

References

[1] W. C. Wong, and A. Fu, "Finding Structure and Characteristics of Web Documents for Classification," in *Proceedings of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (DMKD), Dallas, TX., USA, 2000, pp. 96-105.

[2] Fürnkranz J., "Exploiting Structural Information for Text Classification on the WWW," in *Proceedings of the 3rd Symposium on Intelligent Data Analysis* (IDA-99), Springer-Verlag, 1999, pp. 487-497.

[3] Soderland. S., "Learning to extract text-based information from the World Wide Web," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (KDD 97), Newport Beach, CA., 1997, pp. 251-254.

[4] S. H. Lin, M. C. Chen, J. M. Ho, and Y. M. Huang, "{ACIRD}: Intelligent Internet Document Organization and Retrieval," *Journal of Knowledge and Data Engineering*, Vol. 14, No. 3, 2002, pp. 599-614.

[5] H. J. Oh, S. H. Myaeng, and M. H. Lee, "A practical hypertext categorization method using links and incrementally available class information," in *Proceedings of the 23rd ACM SIGIR 2000 Conference*, Athenes, Greece, July 2000, pp. 264-271.

[6] Y. H. Kuo, and M. H. Wong, "Web Document Classification based on Hyperlinks and Document Semantics," in *Proceedings of PRICAI 2000 Workshop on Text and Web Mining*, Aug. 2000, pp. 44-51.

[7] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," in *Proceedings of ACM-SIGMOD*, 1998, pp. 307-318.

[8] Mark Craven, "Using statistical and relational methods to characterize hyperlink paths," in *Proceedings of Artificial Intelligence and Link Analysis* (AAAI), Fall Symposium, Orlando, Florida, 1998, pp. 14-20.

[9] B. Y. Ricardo, and R. N. Berthier, *Modern Information Retrieval*, Chapter 2, ACM press, 1999.