Automatic Wrapper Generation and Maintenance

Yingju Xia, Yuhang Yang, Fujiang Ge, Shu Zhang, Hao Yu

Fujitsu Research & Development Center Co.,LTD.
15F Tower A, Ocean International Center, No.56 Dong Si Huan Zhong Rd, Chaoyang District, Beijing, China ,100025
yjxia@cn.fujitsu.com

Abstract. This paper investigates automatic wrapper generation and maintenance for Forums, Blogs and News web sites. Web pages are increasingly dynamically generated using a common template populated with data from databases. This paper proposes a novel method that uses tree alignment and transfer learning method to generate the wrapper from this kind of web pages. The tree alignment algorithm is adopted to find the best matching structure of the input web pages. A kind of linear regression method is employed to get the weight of different tag-matching. A transfer learning method is adopted to find the most likely content block. A wrapper built on the most probable content block and the repeating patterns extracts data from web pages. The wrapper maintenance arises because web source may experiment changes that invalidate the current wrappers. This paper presents a wrapper maintenance method using a log likelihood ratio test for detecting the change points on the similarity series which gotten from the wrapper and input web pages. The wrapper generation method is applied to generate a wrapper once the web source change is detected. Experimental results show that the method achieves high accuracy and has steady performance

Keywords: Data extraction, Wrapper generation, Wrapper maintenance.

1 Introduction

Web-based information is typically formatted to be read by human users, not by computer applications. Information agents are being proposed which automatically extract information from multiple websites. Data is typically extracted from web sources by writing specialized programs, called wrappers (Laender et al. 2002), which identify data of interest and map them to a suitable format.

Many approaches have been reported in the literature for wrapper generation. Detailed discussions of various approaches can be found in several surveys (Chang et al., 2006; Laender et al., 2002).

Early approaches were based on manual techniques (Atzeni and Mecca, 1997; Crescenzi and Mecca, 1998; Huck et al., 1998; Sahuguet and Azavant, 1999). By observing a web page and its source code, the programmer find some patterns from the page and then write a program to extract data from the web pages. A key problem with manually coded wrappers is that writing them is a difficult and labor-intensive task, and tends to be brittle and difficult to maintain.

Other approaches have some degrees of automation. In semi-automatic approaches (Cohen et al., 2002; Irmak and Suel, 2006; Kushmerick, 2000; Muslea et al., 1999; Pinto et al., 2003; Wang and Hu, 2002; Zheng et al., 2007), a set of extraction rules are learnt from a set of manually labeled pages or data records. These rules are then used to extract data items from similar pages. This method still requires substantial manual efforts.

In automatic methods, Arasu and Hector (2003), Chang and Liu (2001) and Crescenzi et al. (2001) found patterns or grammars from multiple pages containing similar data records. Wang and Lochovsky (2003) treated the input pages as strings and employed an algorithm to discover the continuously repeated substrings using suffix trees. Lerman et al. (2004) utilized the detailed data in the page behind the current page to identify data records. Simon and Lausen (2005) identified and ranked potential repeated patterns using visual features. Then matched

subsequences of the pattern with the highest weight was aligned with global multiple sequence alignment techniques.

Several methods were presented to address the wrapper maintenance problem. Kushmerick (1999) defined a problem called "wrapper verification," which checks if a wrapper stops extracting correct data. Their proposed solution analyzes pages and extracted information, and detects the page changes. If the pages have changed, the designer is notified, so that she can re-learn the wrapper from the pages with the new structure. Lerman et al. (2003) developed a method for repairing wrappers in the case of small mark-up changes. Chidlovskii (2001) presented an automatic maintenance approach to repairing wrappers under the assumption that there are only small changes. Raposo et al.(2005) made wrappers collect some results from valid queries during their operation, and when the source changes, use those results to generate a new training set of labeled examples to bootstrap the wrapper induction process again. Meng et al. (2003) presented a schema-guided approach which is based on the observation that despite various page changes, many important features of the pages are preserved, such as syntactic patterns, annotations, and hyperlinks of the extracted data items. Their approach uses these preserved features to identify the locations of the desired values in the changed pages, and repair wrappers correspondingly by inducing semantic blocks from the HTML tree.

Those previous methods focused on the list pages (Each of such pages contains lists of objects, for example, the pages in the shopping website such as Amazon.com.). This kind of web pages can be retrieved using queries which enable the "wrapper verification" procedure. But for the web pages from News, Forum and Blogs, the "wrapper verification" approach cannot be utilized because these web pages cannot be retrieved using queries and such no valid training set can be provided for the wrapper maintenance.

This paper presents a method that uses tree alignment to automatically build wrapper from web pages coming from News, Forums and Blogs websites. A kind of linear regression method is proposed to get the weight of different tag-matching. Based on the alignment, we merge the trees into one union tree whose nodes record the statistical information gotten from multiple web pages. We use a transfer learning method to find the most likely content block and use the alignment algorithm to detect the repeat patterns on the union tree. A log likelihood ratio test is adopted to the wrapper maintenance. Because the likelihood ratios describe evidence rather than embody a decision, they can easily be adapted to the various goals for which inferential statistics might be used. The likelihood ratios provide an intuitive approach to summarizing the evidence provided by an experiment in wrapper maintenance scenario.

2 Related Works

For the wrapper generation, in the sense of techniques used, the most relevant approaches are (Zhai and Liu, 2005; Zigoris et al., 2006).

Zhai and Liu(2005) used partial alignment method to align and extract data items from the identified data records. Zigoris et al.(2006) used Support Vector Machines (SVM) for learning the tree alignment parameters. With well-tuned parameters these models are resilient.

Compared with these methods, the wrapper generation method proposed in this study presents a kind of linear regression method to get the weight of different tag-matching. The algorithm is dedicated to adopt different node features and different matching weights while the others haven't take into account the categories of html tags, the properties of different level nodes and the text features.

Another major difference between the proposed method and the previous works is the way the alignment algorithm and the statistics are used. Zhai and Liu(2005) used the alignment algorithm to align the data items (data fields) from the identified data records. A link was created when a matching was found. The method proposed in this paper utilizes the alignment algorithm to obtain skeleton of the input trees, merges the trees into one union tree and records the statistical information. The proposed method employed the most probable content block finding step to locate the content blocks. The statistics recorded in the union tree makes this step

more accurate because the heuristic is often used to differentiate the content from junk information.

We present a transfer learning method to get the weight of each feature when finding the most probable content block. The proposed method gets steady performance due to the statistics used.

For the wrapper maintenance, as mentioned in Section 1, the previous methods focused on the list pages. The "wrapper verification" approach cannot be utilized on the web pages from News, Forums and Blogs. There are no literature considering the wrapper maintenance on News, Forums and Blogs websites. In this paper, a log likelihood ratio test is adopted to wrapper maintenance.

3 Wrapper Generation

There are several steps in the proposed method.

(1) Wrapper generation. We use the tree alignment methods to calculate the similarity between input web pages and build a wrapper on tree alignment results. The tree alignment method is also used to calculate the similarity between wrapper and the input web pages. The input trees are merged into one union tree whose nodes record the statistical information such as the times a node has been aligned, the text length of the node. A heuristic method is employed to find the most probable content block. The alignment algorithm is utilized again to detect the repeating patterns on the union tree. The wrapper is generated based on the most probable content block and the repeating patterns.

(2) Similarity series generation. A similarity series was built by calculating the similarity between the input web pages and the current wrapper using the tree alignment algorithm proposed in this paper. The similarity series is in the order of the input web pages' timestamp.

(3) Change point detection and wrapper regeneration. A log likelihood ratio test is utilized to detect the change points on the similarity series. The wrapper generation method is applied again to generate a wrapper once a change point is detected.

In this study, we are interested in one specific type of tree called labeled ordered rooted tree. A rooted tree is a tree whose root vertex is fixed. Ordered rooted trees are rooted trees in which the relative order of the children is fixed for each vertex. We use the tree edit distance to evaluate the structural similarities between Web pages. In its traditional formulation, the tree edit distance problem considers three operations: node removal, node insertion and node replacement. The solution of this problem consists in determining the minimal set of operations to transform one tree into another. Another equivalent formulation of this problem is to discover a mapping with minimum cost between the two trees.

In this work, we focus on setting the weight (cost) of different node mapping (tag-matching). One of the major contributions of our work is a kind of linear regression method for getting the weight of different tag-matching. Another contribution of our work is the way we use the similarity between trees and the transfer learning method which is used for finding the most likely content block.

3.1 Automatically getting tag-matching weight

The main problem of the previous method is that they did not consider about employing different weights for various tag-matching. For example, the HTML tags are divided into two categories: block elements and inline elements. The block elements are elements that usually, but not always, contain other elements. They normally act as containers of some sort. The inline elements normally mark up the semantic meaning of something. Furthermore, the level of the different nodes should also be considered. The higher-level nodes should have higher weight as the higher-level nodes usually act as bigger structure block. Different weight should be assigned to different type of tag-matching.

In this study, a kind of linear regression method is employed to get the weight of different tag-matching. First, we found a collection of similar web pages belong to the same "class" (The

web pages share the common format and layout characteristics, usually generated with the same template, for example, the web pages of the same board in one Forum website). It's feasible to get this kind of web pages collection automatically. Next, we will use this web pages collection for getting the optimal weighting schema.

Let w_i be the weight of tag-matching and $w_i > w_j$ for $i < j$. Let D_{mn} be the sum of the gains in the best alignment between the trees T_m and T_n. $D_{mn} = \sum_i w_i t_i^{mn}$ (1)

Where t_i^{mn} is the number of w_i occur in the alignment procedure.

The sum of the gains in the collection is:

$$f = \sum_{m,n} D_{mn} = \sum_{m,n} \sum_i w_i t_i^{mn} = \sum_i w_i \sum_{m,n} t_i^{mn} \quad (2)$$

Because the collection is the similar web pages belonging to the same "class", a set of w_i is selected which makes the maximum f.

To get $argmax_w \sum_i w_i \sum_{m,n} t_i^{mn}$, a constraint $\sum_i w_i^2 = 1$ is added. The group of equations is rewritten as: $f = \sum_i w_i C_i + \lambda\left(\sum_i w_i^2 - 1\right),\ C_i = \sum_{m,n} t_i^{mn},\ \sum_i w_i^2 = 1$ (3)

The solution of the above equations is used as the weight of each type of tag-matching (w_i).

Figure 1 illustrates an example of the weight setting method. For one collection of similar web pages belong to the same "class", we calculate the sum of the alignment gains (or the similarity) for each weighting schema. The best weighting schema is the one maximize the sum of the gains. That means to find a set of w_i that output the maximum f in the equations (3).

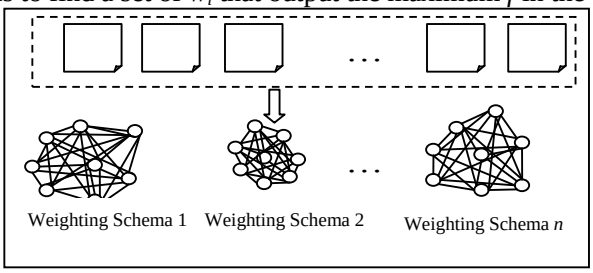

Figure 1: An example of weight setting

3.2 Transfer learning method for the most probable content block detecting

Using the alignment algorithm, we can find whether a node has been aligned. We then merge the two trees into one union tree and record the alignment information in each node. After processing several trees, we can use the statistic such as the ratio of the times a node has been aligned to make the decision whether this node should be kept or not. The union tree becomes more compact after deleting some useless nodes.

The next step is finding the most probable content block (the data in the content block is what we want to extract). In general cases, there is one content block in news page and several content blocks in forum and blog page. The content block detecting method is shown below:

$$\text{score} = \sum_i w_i f_i \quad (4)$$

Where, f_i is the feature and w_i is its weight.

There are many heuristic features (Christian , 2009) such as the variance of the text length of nodes, the ratio of the length of the link to the length of the text in the node, the ratio of the fixed text length and number of stop words inside the DOM node.

The remained issue is how to get the weight of each feature. Since there are three related types: News, Forums and Blogs. We can consider this problem as transfer learning (Jing, 2009). We are interested in getting the weight of target webpage type T and we have labeled instance for K auxiliary type $A_1, \ldots, A_k$. Let w^k denote the weight vector of the linear classifier for the auxiliary type A_k and w^T denote the weight vector for the target type T. we now assume that these weight vectors are related through a common component v:

$$\omega^T = \mu^T + v\ ,\ \omega^k = \mu^k + v, \quad \text{for } k = 1,2, \ldots, K \quad (5)$$

If we assume that only weight of certain general features can be shared between different web page types, we can force certain dimensions of v to be 0. We use a square matrix F and set Fv=0. The entries of F are set to *0* except that $F_{i,i}$=1 if we want to force v_i=0.

Now we can learn these weight vectors in a transfer learning framework. Let x represents the feature vector of a candidate web page, and y ∈ {+1, -1} represent a class label. Let $D_T = \{(x_i^T, y_i^T)\}_{i=1}^{N_T}$ denote the set of labeled instances for the target type T. Let $D_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ denotes the labeled instance for the auxiliary type A_k.

We learn the optimal weight vectors $\{\hat{\mu}^k\}_{k=1}^K, \hat{\mu}^T$ and $\hat{v}$ by optimizing the following objective function:

$$(\{\hat{\mu}^k\}_{k=1}^K, \hat{\mu}^T, \hat{v}) = \arg\min_{\{\mu^k\}, \mu^T, v, Fv=0} [L(D_T, \mu^T + v) + \sum_{k=1}^K L(D_k, \mu^k + v) + \lambda_\mu^T \|\mu^T\|^2 + \sum_{k=1}^K \lambda_\mu^k \|\mu^k\|^2 + \lambda_v \|v\|^2] \quad (6)$$

Once we get the most probable content, we use the alignment algorithm to find the repeat patterns. We first split the union tree into several subtrees according to the content block nodes. The alignment algorithm is used to measure the similarity between subtrees. Also in the alignment, we consider about node's weight according to the level and category. This step is especially useful for the web pages coming from Forums and Blogs. In the News web pages, the content block itself is usually used as the extracting pattern.

Thus, by alignment, merging, finding content block and mining the repeat patterns, we can get a wrapper to extract the data from web pages.

4 Wrapper Maintenance

The wrapper maintenance arises because the template of the web source may experiment changes that invalidate the current wrappers.

Figure 2 shows an example of the template change detection. The x-axis shows the web pages of one website ordered by the timestamp. Let *time*(*i*) be the time of the webpage *i*, then *time*(*i*)<*time*(*j*) for *i*<*j*. The y-axis shows the similarities between the current wrapper and the input web pages. The similarities are calculated using the tree alignment algorithm presented in this paper. The website's template changed at *tc* which causes the low similarities between the current wrapper and the web pages after *tc*. This means that the wrapper should also change due to the change of the website's template. In this scenario, the wrapper maintenance includes two steps. The first step is the detection of the change points in the similarity series. The second step is repairing or re-generating the wrapper using the web pages after *tc*.

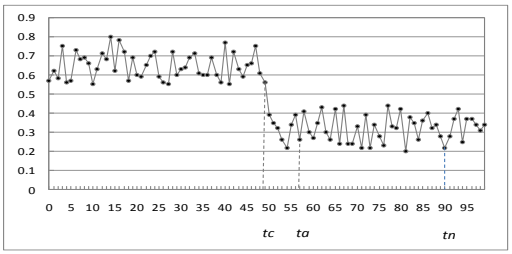

Figure 2: An example of template change detection

We use the log likelihood ratio test (Zeitouni et al., 1992) to detect the change points. Because the likelihood ratios describe evidence rather than embody a decision, they can easily be adapted to the various goals for which inferential statistics might be used. The likelihood ratios provide an intuitive approach to summarizing the evidence provided by an experiment in wrapper maintenance scenario. Let p be the actual distribution of the similarities series.

p_{θ_0} be the distribution under H_0 $p_{\theta 1}$ be the distribution under H_1

We use y_i to denote the similarity between the *i-th* web page and the current wrapper. Let us introduce the following hypotheses:

H_0: $p(y_i \mid y_{i-1}, \cdots, y_1) = p_{\theta_0}(y_i \mid y_{i-1}, \cdots, y_1)$

H_1: There is a time t_c such that

For $1 \le i \le t_c - 1$: $p(y_i \mid y_{i-1}, \cdots, y_1) = p_{\theta_0}(y_i \mid y_{i-1}, \cdots, y_1)$

For $t_c \le i \le k$: $p(y_i \mid y_{i-1}, \cdots, y_{t_c}) = p_{\theta_1}(y_i \mid y_{i-1}, \cdots, y_{t_c})$ (7)

We use the log of likelihood ratio $S_i = ln\frac{L_{\theta_1}(y_i)}{L_{\theta_0}(y_i)}$ to indicate the relative probability of the data.

S_j^k denotes the sum of log likelihood ratio. $S_j^k = \sum_{i=j}^{k} S_i$ (8)

To simplify, suppose that y_i are independent and normally distributed with common variance σ^2. Consider $y_i \sim N(\mu, \sigma^2)$.

$\mu = \mu_0$ under H_0 $\mu = \mu_1$ under H_1

We get $S_j^k = \frac{b}{\sigma}\sum_{i=j}^{k}(y_i - \mu_0 - \frac{v}{2})$ Where: $v = \mu_1 - \mu_0, b = \frac{\mu_1 - \mu_0}{\sigma}$

Hence, we get the following hypotheses:

H_0: $S_j^k \sim N(-\frac{mv^2}{2\sigma^2}, mb^2)$ H_1: $S_j^k \sim N(\frac{mv^2}{2\sigma^2}, mb^2)$ (9)

The problem becomes testing whether S_j^k is $N(-\frac{mv^2}{2\sigma^2}, mb^2)$. If H_1 holds, a change point is reported at *j-th* web pages. Take Figure 6 as an example, we use a window (size: m=(*tn*-*ta*)), and calculate the S_j^{j+m}. The H_1 holds at *i+m* will raise an alarm at *i.* for example, the H_1 holds at *tn* and raises an alarm at *ta.* There is (*ta*-*tc*) delay since the change point actually occurs at *tc*. In this paper, different window sizes were set to evaluate the change point detection performance.

As shown above, we assume *yi* are independent and normally distributed with common variance σ^2. This tends to be less flexible since it relies on a model assumption. In the community of statistics, some non-parametric density estimation is used for calculating the likelihood ratio (Brodsky and Darkhovsky, 1993). However, non-parametric density estimation is known to be a hard problem (Hardel et al. 2004; Huang et al., 2007), it may not be promising in the practice. The experimental results in Section 3 have shown this. One way to alleviating this difficulty is to directly estimate the ratio of probability densities, not the probability densities themselves. Recently, direct density-ratio estimation has been actively explored in the machine learning and the Kullback Leibler Importance Estimation Procedure (KLIEP)(Sugiyama et al., 2008). We also use the KLIEP and give the experimental results in Section 5.

5 Experimental Results

5.1 Evaluation of Wrapper Generation

The wrapper generation approach is compared with the Zhai and Liu(2005) (PA in Table 1) and Zigoris et al.(2006) (SVM in Table 1). For each web page category (News, Forum and Blog), 40 websites and 6000 web pages were selected for the experiments.

The experimental results are shown in Table 1. PA performs better in the news websites than in the forum and blog while our method gets the best performance in blogs. It's because that PA method is more suitable for news websites. The experimental results also showed that the transfer learning for tree alignment parameters outperform the SVM method.

We evaluated the system performance under different size of union tree. Figure 3 shows the experimental results. Here the 'news_p' and 'news_r' are the precision and recall gotten from the news corpus respectively, and 'forum_p' and 'forum_r' are the precision and recall gotten from the forum corpus respectively. The x axis shows number of the trees merged into the union tree. We can see that the system tend to convergence while the number is about 30. That means

we can use about 30 samples to build the union tree and to get the extraction template, which will save much time.

We evaluated the impact of the size of the collection for getting the tag-matching weight. From the Figure 4, we can see that the system performance better on the blog corpus again. We should notice that the weight setting procedure needs more samples than the union tree building (50 web pages in the forum and blog corpus, 80 web pages in the news corpus). Since the weight setting is time-consuming (C_n^2 tree alignment on the collection of n web pages), we think it's better to get them in advance.

We also evaluated the annotated samples needed for the transfer learning method. The experiment results are shown in Figure 5. We can see that the more the annotated samples, the better the system performance. The system tends to convergence while the number of the annotated sample is about 200.

Table 1: The experimental results

		PA	SVM	Our method
News	Precision	0.912	0.892	**0.903**
	Recall	0.865	0.933	**0.956**
Forum	Precision	0.845	0.918	**0.932**
	Recall	0.891	0.946	**0.965**
Blog	Precision	0.848	0.921	**0.941**
	Recall	0.903	0.958	**0.969**

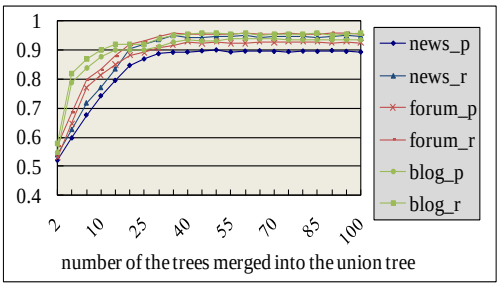

Figure 3: Impact of number of trees merged into the union tree

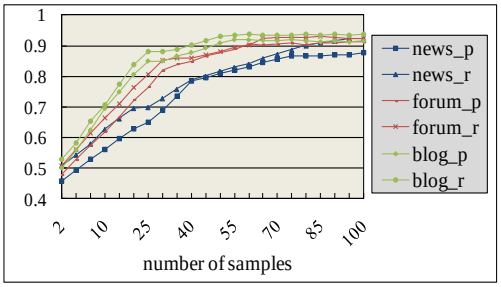

Figure 4: Impact of the number of samples for getting the tag-matching weight

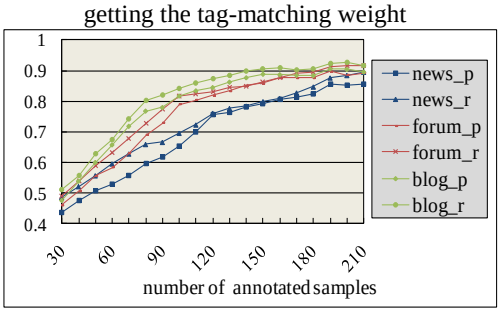

Figure 5: Impact of the number of annotated samples for transfer learning

5.2 Evaluation of Wrapper Maintenance

In order to evaluate the effectiveness of the proposed wrapper maintenance approach, we monitored a set of websites for several months. Our experiment was designed to model the scenario in which the goal is to detect whether a site's template has changed. For a fixed site, we got a sequence (p_1, p_2, ..., p_n) of gathered pages. The wrapper maintenance system should return TRUE iff the site's template changes at p_i which means $w_{i-1} \neq w_i$. Where the w_i is the wrapper for p_i. We measured the performance in term of 2*2 matrix shown in Table 2.

Table 2: *Wrapper maintenance evaluation matrix*

	$w_{i-1}=w_i$	$w_{i-1}\neq w_i$
Predict TRUE	*n1*	*n2*
Predict FALSE	*n3*	*n4*

Several metrics are derived from this matrix.

$$Accuracy = \frac{n2+n3}{n1+n2+n3+n4} \tag{10}$$

$$Precision = \frac{n2}{n1+n2} \qquad Recall = \frac{n2}{n2+n4} \qquad F = \frac{2*Recall*Precision}{Recall+Precision} \tag{11}$$

We manually made some data for wrapper maintenance evaluation because the real data is not enough (26, 30 and 35 for News, Forums and Blogs respectively). We made some test data manually. We change the template of website and put data in it to make some new pages. These kinds of data make the wrapper maintenance problem more difficult. We got 100 data sets for each type of websites (News, Forums and Blogs) including the real data gathered. Experimental results are shown in Figure 6. We can see that the performance on the merged test set (News_M, Forum_M, Blog_M) was worse than that on the real test set.

As shown in Section 4, there are parametric and non-parametric density estimation method and direct density-ratio estimation method for the log likelihood ration test. We compared these methods and got the experimental results shown in the Figure 7, 8 and 9. Here, the 'LLR-Norm' means the parametric method using normal distribution, 'LLR-Non' means the non-parametric method and 'KLIEP' means the direct density-ratio estimation method shown in (Sugiyama et al., 2008). The experimental results show that the LLR-Norm method outperforms the LLR-Non and KLIEP. As mentioned in Section 4, non-parametric density estimation is known to be a hard problem (Hardel et al. 2004; Huang et al., 2007), it may not be promising in the practice. The reason why the KLIEP is worse than LLR-Norm is that it is a batch algorithm and not suitable for the change point detection. The experiments follow the principles of Occam's Razor. In many cases, the simplest solution is usually the correct one.

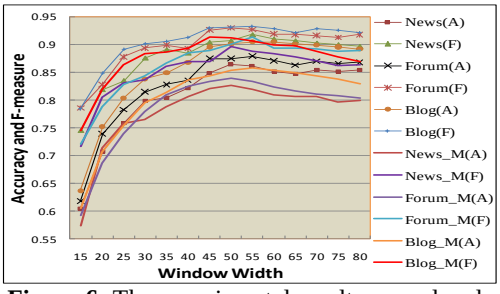

Figure 6: The experimental results on real and merged test set

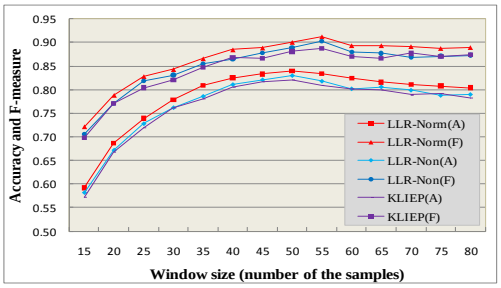

Figure 8: Experimental results on Forums

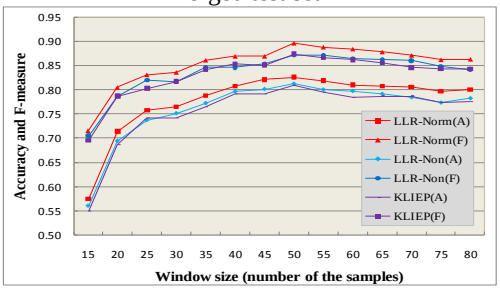

Figure 7: Experimental results on News websites

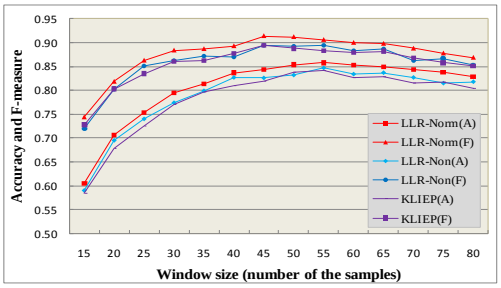

Figure 9: Experimental results on Blogs

6 Conclusions and Future Works

In this work, a method that uses tree alignment and transfer learning method is proposed to generate the wrapper from web pages of Forums, Blogs and News web sites. The tree alignment algorithm is adopted to find the best matching structure of the input web pages. A kind of linear regression method is employed to get the weight of different tag-matching. Based on the alignment, we merge the trees into one union tree whose nodes record the statistical information gotten from multiple web pages. We use a transfer learning method to find the most likely content block and use the alignment algorithm to detect the repeat patterns on the union tree. In the wrapper maintenance approach, the tree alignment algorithm is utilized as a metric of the similarity between wrapper and web pages. A log likelihood ratio test is adopted to detect the change points on the similarity series. The wrapper generation method is applied again to

generate a wrapper once the web source change is detected. This joint wrapper generation and maintenance method can be applied to the kind of websites whose web pages are dynamically generated using a common template populated with data from databases. Experimental results show that the method achieves high accuracy and has steady performance.

There are several important issues remaining to be addressed in the future work. It is important to make the most probable content block detecting more accurately and thus enhance the repeating pattern mining procedure. In wrapper maintenance problem, it's a challenge to explore using fewer samples.

References

Arasu A and Hector GM. 2003. Extracting structured data from Web pages. Proceedings of the 2003 ACM SIGMOD international conference. San Diego, California. PP337-348.

Atzeni P and Mecca G. 1997. Cut and Paste. Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium. Tucson, Arizona, United States. PP144-153.

Chang C. and Lui SL. 2001. IEPAD: Information extraction based on pattern discovery. Proceedings of the 10th World Wide Web. Hong Kong. PP681-688.

Brodsky B and Darkhovsky B. Nonparametric Methods in Change-Point Problems, Kluwer Academic Publishers, 1993.

Chang CH, Kayed M, Girgis MR, Shaalan K. 2006. A survey of Web information extraction systems. IEEE Transactions On Knowledge and Data Engineering, 2006, pp1411-1428

Chidlovskii B. 2001. Automatic repairing of web wrap-pers. Proceedings of the 3rd international workshop on Web information and data management. Atlanta, Georgia, USA, 9 November 2001, PP24-30

Cohen W, Hurst M, and Jensen L. 2002. A flexible learning system for wrapping tables and lists in HTML documents. Proceedings of the 11th inter-national conference on World Wide Web. Honolulu, Hawaii, USA. PP232-241.

Crescenzi V and Mecca G. 1998. Grammars have ex-ceptions. Information Systems, 23(8), PP539-565

Crescenzi V, Mecca G and Merialdo P. 2001. Roadrunner: Towards automatic data extraction from large web sites. In proceedings of the 26th International Conference on Very Large Database Systems. Rome, Italy, 2001. PP 109-118.

Christian K. 2009. A Densitometric Analysis of Web Template Content. WWW 2009, April 20–24, 2009, Madrid, Spain. PP 1165-1166

Hardel W, Muller M, Sperlich S and Werwatz A. Non-parametric and Semi-parametric Models, Springer Series in Statistics, Springer, Berlin, 2004

Huck G, Frankhause P, Aberer K and Neuhold E J. 1998. Jedi: extracting and synthesizing information from the web. In Proceedings of 3rd IFCIS International Conference on Cooperative Information Systems. New York, USA, 20-22 Aug 1998. PP32-41.

Huang J, Smola A, Gretton A, Borgwardt K and Schol-kopf B, Correcting Sample Selection Bias by Unla-beled Data, In Advance in Neural Information Processing Systems, Vol. 19, MiT press, Cambridge, MA, 2007

Irmak U and Suel T. 2006. Interactive wrapper genera-tion with minimal user effort. In Proceedings of the 15th World Wide Web. Scotland, May 23-26, 2006. PP553-563.

Jing J. 2009. Multi-task Transfer Learning for Weakly-Supervised Relation Extraction. The 47 th Association for Computational Linguistics, August 2009, pp1012-1020.

Keogh E, Chu S, Hart D, and Pazzani M. 2003. Seg-menting time series: A survey and novel approach. In Data Mining in Time Series Databases. second ed. World Scientific, 2003.

Kushmerick N. 1999, Regression testing for wrapper maintenance. In Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-1999), 1999, PP74-79

Kushmerick N. 2000. Wrapper induction: efficiency and expressiveness. Artificial Intelligence, 118(1-2), April 2000, PP15-68.

Laender AHF, Ribeiro Neto BA, da Silva AD, Teixeira JS, 2002. A Brief Survey of Web Data Extraction Tools. SIGMOD Record, 31(2), June 2002. pp84-93

Lerman K, Minton S N, Knoblock C A. 2003. Wrapper Maintenance: A Machine Learning Approach. Jour-nal of Artificial Intelligence Research 18(2003), PP149-181

Lerman K, Getoor L, Minton S and Knoblock C. 2004. Using the Structure of Web Sites for Automatic Segmentation of Tables. In Proceedings of the 2004 ACM SIGMOD international conference on Man-agement of data. Paris, France, 2004. PP119-130

Meng XF, Hu D D, Li C. 2003. Schema-Guided Wrap-per Maintenance for Web-Data Extraction, Proceedings of the 5th ACM international workshop on Web information and data management, November 7-8, 2003, New Orleans, Louisiana, USA.,PP1-8

Muslea I, Minton S and Knoblock C. 1999. A hierarchical approach to wrapper induction. In Proceedings of the third annual conference on Autonomous Agents, Seattle, Washington, United States, 1999. PP190-197.

Pinto D, McCallum A, Wei X and Bruce W. 2003. Table Extraction Using Conditional Random Fields. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. Toronto, Canada, 2003. PP235-242.

Raposo J, Pan A, Álvarez M , Hidalgo J. 2005. Automatically maintaining wrappers for semi-structured web sources. Proceedings of the 9th International Database Engineering & Application Symposium. 25-27 July 2005, Montreal, Canada. PP 105- 114

Sahuguet A and Azavant F. 1999. Web ecology: Re-cycling HTML pages as XML documents suing W4F. In Proceedings of ACM SIGMOD Workshop, Pennsylvania, June 3-4, 1999.

Simon K and Lausen G. 2005. ViPER: Augmenting automatic information extraction with visual perceptions. In Proceedings of the 14th ACM international conference on Information and knowledge management. Bremen, Germany, 2005. pp 381-388.

Sugiyama M, Suzuki T, nakajima S, Kashima H, Von Bunau P and Kawanabe M. Direct Importance Estimation for Covariance Shift Adaptation, Annals of the Institute of Statistical Mathematics, 60(4), 2008

Wang JY, Lochovsky FH. 2003.Data extraction and label assignment for Web databases. In Proceedings of the 12th World Wide Web. Budapest,Hungary, 2003. PP187-196.

Wang Y and Hu J. 2002. A machine learning based approach for table detection on the Web. In Proceed-ings of the 11th World Wide Web. Honolulu, Hawaii, USA, 2002. PP242-250.

Zeitouni O, Ziv J and Merhav N. 1992. When Is the Generalized Likelihood Ration Test Optimal? IEEE Transactions on Information Theory, VOL 38, No. 5, PP1597-1602

Zhai YH and Liu B. 2005. Web data extraction based on partial tree alignment. In Proceedings of the 14th World Wide Web. Chiba, Japan, May 10-14, 2005, PP76-85.

Zhao HK, Meng WY, Wu ZH, Raghavan V and Yu C. 2005. Fully Automatic Wrapper Generation For Search Engines. In Proceedings of the 14th World Wide Web. Chiba, Japan, May 10-14, 2005, PP66-75

Zheng SH Y, Wu D, Song R H, Wen J R. 2007. Joint Optimization of Wrapper Generation and Template Detection. KDD 2007, August 12-15, 2007, San Jose, California USA, PP894-902

Zigoris P, Eads D, Zhang Y. 2006. Unsupervised Learning of Tree Alignment Models for Information Extraction. Sixth IEEE International Conference on Data Mining - Workshops 2006, PP.45-49