Towards Automatic Content Transfer Between Web Pages

Yonghao Long
School of Data and Computer Science, National Engineering Research Center of Digital Life, Sun Yat-san University Guangzhou China
longyh3@mail2.sysu.edu.cn

Xiangping Chen*
Institute of Advanced Technology, Sun Yat-san University, Guangdong Province Engineering Technology Research Center for Mathematical Educational Software, Guangzhou, China
chenxp8@mail.sysu.edu.cn

Yongsheng Rao
School of Computer Science and Educational Software, Guangzhou University, School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
rysheng@163.com

Xiaohong Shi
School of Data and Computer Science, Sun Yat-san University Guangzhou China Research Institute of Sun Yat-sen University in Shenzhen, Shenzhen,China
gz-sxh@163.com

***Abstract*—The Internet provides numerous web pages for learning and reusing. But reusing these pages requires manually modification and integration of the source codes, which is inefficient and complex. In this paper, we propose a method to do the content transfer between web pages. This method is based on the machine-learning, and it can use any web to be the templates. With transferring the contents from source to the reference, a web page is synthesized with the source's contents and reference's styles. The result can help inspiring the designers and reducing the developing cost.**

Keywords: web design; machine learning; web retargeting

I. INTRODUCTION

Generating a satisfying and attractive User Interface(UI) is very important in software development. Well-designed UI will give a good impression to the users and enhance the user experience[1], the purchase intention, and user loyalty[2,3].

High quality UI requires elaborate designing on the layouts and colors, which makes high demands of the designing experience. Designers often seek the inspirations from examples such as tutorials, existing products, and designing books. Utilizing these examples requires the users to forage and learn the useful parts in the examples, and modifying the parts to meet the requirements. The process is time-consuming and error-prone.

Some works were proposed to simplify UI generation, one of the effective solutions is providing reusable knowledge as templates[4]. By filling the contents in a pre-defined standardized structure, the users can quickly get an implementation of UI for use. However, the number of templates is much less than the types of websites. Getting a satisfying template requires a long time foraging and comparing. In addition, the results may need some refinements to be the product. The lack of the template's diversity limits the utilizing of the templates.

The web pages in the Internet provides numerous UIs, a considerable number of them have excellent appearances. If any of them could be a design template, the cost of designing will be reduced significantly. However, the process to manually analyzing and modifying existing UI is very inefficient. Designers need to understand the reference page firstly. Then, they extract the effective parts in the reference, and integrate them in the original web page. Some codes and contents in these parts need to be modified during the integrating. These procedures are time-consuming and error-prone.

Our study aims to automate the process of transforming the contents between two web pages. We firstly extract all the elements in the example and source pages. Because there are many similar elements in a page which are used to exhibit related contents, our approach clusters the elements. After that, we define a mapping between the clusters in the source and the target page. Machine learning method is used in the clustering and matching.

The reminder of the paper is structured as follows, we firstly introduce some works related; and then describe the clustering and matching algorithm. Finally, we discuss results and future work.

II. RELATED WORK

This work draws on two main areas of prior work: the analyzing and modifying of the elements in web and the tools aid in refining the web design.

The reusing of examples requires the adaptation and modification of the source products, which is a time-consuming work for designers. A number of solutions are proposed to simplify this process.

Some studies considered that a design galleries can give the users multiple alternatives for inspiration and gain intuitions about the designing[5]. Some tools are developed in these works[6-8] to provide multiple examples with different ranking strategies. These tools can reduce the cost of foraging and comparing the examples, but the adapting the structures of examples to the target is still hand-coded.

The templates for reusing the design are proposed to automate the reusing of examples [9]. The template provides some standardized predesigned layouts, the users just need

* Corresponding author.

978-1-5090-4400-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDH.2016.29

to fill some contents in them and a web page can be generated. This is easy to use and effective, but it limits the customization and creativity.

Due to the lack of customization of using the templates, some works focused on the reusing of existing UIs, that is, any UI can be the template. Thus the generated UI will be various due to the diversity of existing UIs.

Current studies try to achieve this goal with the aid of machine learning[10]. The Bricolage[11] introduced a web page retargeting method with the help of machine learning. It transfers the content of the source page to the examples thus the generated result will have the examples' styles.

Our method focus on the content transformation between web pages, which is like to the works of Bricolage[11]. But in our work, we introduce a clustering algorithm to group the similar elements together instead of using the visual-based web page segmentation method[16]. This strategy will keep the relationship of similar elements after the content transformation.

III. TRANSFERING THE CONTENTS BETWEEN WEB PAGES

Our method starts from extracting all the elements in the example and source pages. But we only do the transformation between the elements with contents. Most of them are the leaf nodes in the cascading tree of the web page. Considering there are many similar elements in a page which represent the similar contents, we cluster these elements. Then we do the matching between the set of clusters. Both the clustering and matching are based on the Random Forest method[17]. Finally, we replace the contents in the example with the corresponding contents extracted from the source elements.

A. Clustering the Elements With Similar Properties

The extracted elements only contain some basic visual and structural properties such as the sizes, positions, tree levels etc. Some elements have very similar properties such as the neighbor elements with similar shapes and sizes. These similar elements often represent the similar contents such as a part of navigation bar or news as the figure 1 shows. We consider that giving these similar elements with same transforming strategy will keep the original structure most and get a robust mapping. This idea has been proved to be effective in our previous work[12].

Figure 1. The similar elements in the web page.

The clustering method needs to analyze the similarities of the elements' properties. Firstly, we extract the properties of the elements with standardized representation. The properties we used in the clustering is described in the table 1.

TABLE I. DIMENSIONS FOR THE CLUSTERING

Dimension Type	Dimension Name
positions	x, y, relativeX, relativeY
sizes	width, height, relative, RelativeH, aspectRatio
types	containImage, containText, tags, selector
tree structures	siblingNumber, childrenNumber, level

These properties are the features of the elements and we use the Random Forest method to learn and predicting whether the two elements can be grouped together or not. We start from manually do the clustering in some pages, and use the results to generating a model, this model will help to predicting whether the two elements can be grouped together.

For a web page with n elements, we do the above predicting on each two of the elements and get an n*n matrix of the predicting result. Then we do the top-down clustering on the elements, the algorithm is described as follows:

Algorithm 1: cluster(e)
Input:
The web's elements *e*, the matrix of predicting result .
Output: the elements with clustering labels.
Description:
The algorithm starts from the root of the web.
For an element *e*, the children of *e* is denoted by *e.children*, and the sibling of *e* is denoted by *e.sibling*, the parent of *e* is *P(e)*, and other elements which belong to the same cluster are denoted by *e.cluster* .

Start:
maxScore = r(e,e);
for $e_s \in e.sibling \cup P(e).cluster.children$ {
  if $(r(e,e_s) \geqslant maxScore*T)$
    then *e* and e_s belong to the same cluster
}
for $e_c \in e.children$
  do cluster(e_c);
End

We set the threshold of T to be 0.8, and the result is relatively stable in a wide range of the threshold's changing.

B. Building Mapping Between the Pages

The clustering step will group the similar elements together, the next is to build a mapping between the web pages. This problem is very similar to the calculation of unordered tree editing distance[13], which has been proven to be an NP-complete problem[14]. Some approximate algorithm[15] had been proposed to solve this problem. We use the Random Forest method to do this work.

We propose a cluster-based matching method which firstly generating the mapping between the clusters in two pages, and then doing matching between the clusters. In this

initial study, we only focus on the leaf nodes of the tree which contain the text and image contents.

Building the mapping of the leaf nodes between the two pages is simpler than doing matching with the whole elements in the pages. There are two obvious advantages in this strategy, the one is only a few of elements need to be considered and the other is the structures of the nodes which are taken into consideration is simpler. And the goal of our study is retargeting the contents that the inner nodes are mostly used for layout and have less effect on the contents.

As same as the similarity-evaluation of elements in the previous section, we use a machine-learning method to aid in the matching. With manually doing some matching as the ground truth, we can get a model to predict whether the two clusters can be matched or not. The features we selected in this step are very similar but the calculations are different. Since the clusters always contain more than one element, the calculation between the two clusters in one dimension will both consider the closest distance and the average distance. We also include the number of elements and the total areas of the cluster into the matching evaluation. This preliminary method has got some reasonable results.

C. *Retargeting The Source Contents to The Example*

The last step is retargeting the source contents to the example and synthesizing a new page. In this paper, we focus on the retargeting of texts and images, which are accomplished with the jQuery.

For the elements which belong to the texts, we firstly delete the tags which will influence the style of the fonts such the <b>, <i>, and <u> etc. After that, the contents in the corresponding elements in the example will be replaced to the source text with the following sentence:

```
$(txt).text("contents from source");
```

For the elements which are the images, we firstly move all the images in the source page to the example. In this process, the original folders won't be changed. And then we will modify the source path of images in the matching elements of the example to the new path in the source page with the following sentence:

```
$(img).attr("src","./sourcePage/source URL");
```

Where the sourceURL is the original source path extracted from the source element.

The above process will change the images and texts in the example thus a synthesizing page can be generated.

IV. IMPLEMENTATION

We are still in the process of generating training data to produce better predictions. But some results got by the preliminary method showed the method is reasonable.

Firstly, we have developed an interface to help manually do the matching and clustering (as figure 2 shows). Participants can directly choose and compare some elements and doing the clustering and matching easily. Moreover, the interface can give an intuitive impression on the performance of our clustering and matching method.

(a) Users click a particular area in the screenshot of the web page

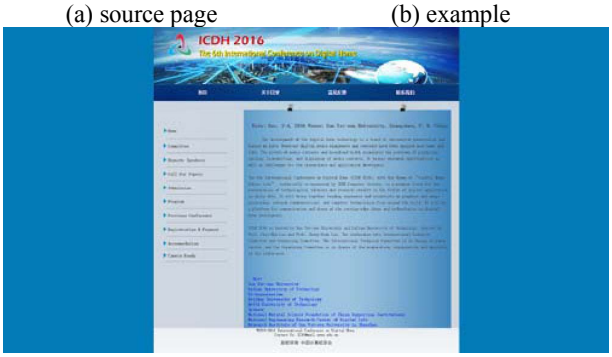
(b) Detailed information of the corresponding element and those elements in the same cluster

Figure 2. The GUI which helps to do clustering and matching.

Secondly, we have done some evaluations on the clustering method and get a satisfying result, preliminary results shows that the clustering result can achieve the accuracy of more than 80%, and the changing of the threshold proposed in the clustering decision won't influence too much on the clustering result.

(a) source page

(b) example



(c) synthesizing page

Figure 3. The similar elements in the web page.

Figure 3 shows the synthesizing page generated in the preliminary method, the generated page will both have the example's appearance and the original structures. But some

source elements are lacked in the synthesizing page, we are testing the transformation of the inner nodes and generate some tags in the spare parts of the elements.

The matching result still needs more training data to produce a better result, and the matching between inner nodes is in the progress, we are also working on the more detailed transformation of the other tags.

V. CONCLUSIONS

This work-in-progress introduced a novel technique for automatically retargeting the content of a web page onto the layout of any other web page. Even novice users can quickly get a new alternative designs for their web pages without restrictions on content. We have observed that our method produces reasonable results, and the original structures can be kept.

We are currently collecting more training data to produce better predictions. After that, we will investigate the matching between all nodes in the web page, and including more sophisticated strategies on the transformation between tags. We also noticed that there are a lot of spare areas which are not utilized; a way of generating some tags in these areas can reduce the number of abandoned elements in the source page. Moreover, from our preliminary observation on the matching result, difference people will generate different matching results; we will investigate the difference of matching between different people and our results.

ACKNOWLEDGMENT

This research is supported by the NSFC Guangdong Joint Fund (No.U1201252), the Science and Technology Planning Project of Guangdong Province (No. 2014B010110003), Educational Commission of Guangdong Province(No. 2013CXZDB001), and the National High-tech R&D Program of China(863 Program, 2015AA015408).

REFERENCES

[1] Flatla, D. R., Reinecke, K., Gutwin, C., & Gajos, K. Z. (2013, April). SPRWeb: Preserving subjective responses to website colour schemes through automatic recolouring. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems pp. 2069-2078

[2] Cyr, D., Head, M., & Larios, H. (2010). Colour appeal in website design within and across cultures: A multi-method evaluation. International journal of human-computer studies, 68(1), 1-21.

[3] Cyr, D.: Modeling Website Design across Cultures: Relationships to Trust, Satisfaction and E-loyalty. Journal of Management Information Systems 24, 47–72 (2008)

[4] Gibson, D., Punera, K., & Tomkins, A. (2005, May). The volume and evolution of web page templates. In Special interest tracks and posters of the 14th international conference on World Wide Web (pp. 830-839). ACM.

[5] Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., ... & Ryall, K. (1997, August). Design galleries: A general approach to setting parameters for computer graphics and animation. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (pp. 389-400). ACM Press/Addison-Wesley Publishing Co.

[6] Herring, S. R., Chang, C. C., Krantzler, J., & Bailey, B. P. (2009, April). Getting inspired!: understanding how and why examples are used in creative design practice. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 87-96). ACM.

[7] Lee, B., Srivastava, S., Kumar, R., Brafman, R., & Klemmer, S. R. (2010, April). Designing with interactive example galleries. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2257-2266). ACM.

[8] Terry, M., & Mynatt, E. D. (2002, October). Side views: persistent, on-demand previews for open-ended tasks. In Proceedings of the 15th annual ACM symposium on User interface software and technology (pp. 71-80). ACM.

[9] Gibson, D., Punera, K., & Tomkins, A. (2005, May). The volume and evolution of web page templates. In Special interest tracks and posters of the 14th international conference on World Wide Web (pp. 830-839). ACM.

[10] Kumar, R., Kim, J., & Klemmer, S. R. (2009, April). Automatic retargeting of web page content. In CHI'09 Extended Abstracts on Human Factors in Computing Systems (pp. 4237-4242). ACM.

[11] Kumar, R., Talton, J. O., Ahmad, S., & Klemmer, S. R. (2011, May). Bricolage: example-based retargeting for web design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2197-2206). ACM.

[12] Chen, X., Long, Y., & Luo, X. (2015). Automatic Color Modification for Web Page Based on Partitional Color Transfer. In Software Reuse for Dynamic Systems in the Cloud and Beyond (pp. 204-220). Springer International Publishing.

[13] Shasha, D., Wang, J. L., Zhang, K., & Shih, F. Y. (1994). Exact and approximate algorithms for unordered tree matching. IEEE Transactions on Systems, Man, and Cybernetics, 24(4), 668-678.

[14] Zhang, K., Statman, R., & Shasha, D. (1992). On the editing distance between unordered labeled trees. Information processing letters, 42(3), 133-139.

[15] Shasha, D., Wang, J. L., Zhang, K., & Shih, F. Y. (1994). Exact and approximate algorithms for unordered tree matching. IEEE Transactions on Systems, Man, and Cybernetics, 24(4), 668-678.

[16] Cai, D., Yu, S., Wen, J. R., & Ma, W. Y. (2003). VIPS: a vision-based page segmentation algorithm (p. 28). Microsoft technical report, MSR-TR-2003-79.

[17] Ho, Tin Kam. "Random decision forests." Proceedings of the Third International Conference on Document Analysis and Recognition, 1995. vol. 1, pp. 278-282. IEEE, 1995.