Attribute Relation Extraction from Template-inconsistent Semi-structured Text by Leveraging Site-level Knowledge

Yang Liu, Fang Liu, Siwei Lai, Kang Liu, Guangyou Zhou, Jun Zhao
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{yang.liu,fliu,swlai,kliu,gyzhou,jzhao}@nlpr.ia.ac.cn

Abstract

A variety of methods have been proposed for attribute-value extraction from semi-structured text with consistent templates (strict semi-text). However, when the templates in semi-structured text are inconsistent (weak semi-text), these methods will work poorly. To overcome the template-inconsistent problem, in this paper, we proposed a novel method to leverage site-level knowledge for attribute-value extraction. First, we use a graph-based random walk model to acquire site-level knowledge. Then we utilize such knowledge to identify weak semi-text in each page and extract attribute-value pairs. The experiments show that, comparing to the baseline method which does not utilize site-level knowledge, our method can improve the extraction performance significantly.

1 Introduction

Among types of relations, attributes (e.g. nationality, date of birth) have emerged as one of the most popular types (Alfonseca et al., 2010), as they capture properties of respective objects (or instances) (e.g. *Kobe Bryant*). Generally, an attribute relation consists of an object, an attribute and its associated value (e.g. *Kobe Bryant - date of birth - August 23, 1978*, where "*August 23, 1978*" is the value of "*date of birth*"). In this paper, we call such a relation an object-attribute-value (OAV) tuple. Many methods have been proposed to extract attributes from semi-structured text (Cafarella et al., 2008)(Venetis et al., 2011)(Crescenzi et al., 2001)(Arasu and Garcia-Molina, 2003) and unstructured text like webpages and Web search query logs (Reisinger and Paşca, 2009)(Paşca et al., 2010)(Pasca and Van Durme, 2007). Semi-structured text (strict semi-text) often has distinctive HTML tags and consistent templates like HTML tables (eg: Wikipedia infoboxes). However, a lot of user-generated semi-structured text with weak structures exist, where their templates generating records are inconsistent and the HTML tags in these templates are less distinctive. In this paper, we focus on the issue of extracting attribute-value (AV) pairs from semi-structured text with inconsistent templates (weak semi-text).

In previous work, Yoshinaga and Torisawa (Yoshinaga and Torisawa, 2007) extracted AV pairs of given objects from semi-structured text. They induced templates via a set of attributes obtained beforehand and used the templates to extract AV pairs. There are two constraints of their method. First, it heavily depends on the initial set of attributes. However, the quality and coverage of the initial set of attributes is hard to control. Second, they hold the assumption that attributes in the same block of semi-structured text are generated with the same template which weak semi-text does not satisfy. Their method mainly concentrated on extraction from semi-structured text with consistent templates (strict semi-text). When facing the weak semi-text with inconsistent templates, it will fail to obtain satisfactory results.

To resolve the problem of inconsistent templates, we propose an unsupervised method by leveraging site-level knowledge to extract AV pairs from weak semi-text. We explore the intrinsic structure connection among pages of the same website to address the problem. We make a two-stage effort: The first stage is to acquire knowledge that reveals the intrinsic similar structures among similar pages of the same site (site-level knowledge); the second stage is to leverage site-level knowledge to assist the AV pair extraction in weak semi-text.

In the paper, we present a novel approach that leverages site-level knowledge to extract instances' attributes and their values from weak semi-text. To the best of our knowledge, little

International Joint Conference on Natural Language Processing, pages 1097–1101,
Nagoya, Japan, 14-18 October 2013.

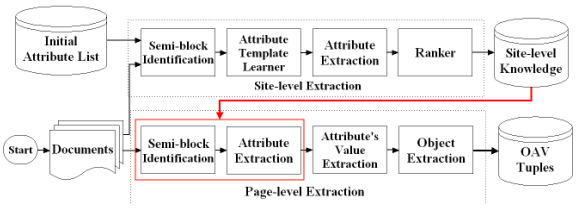

Figure 1: System overview.

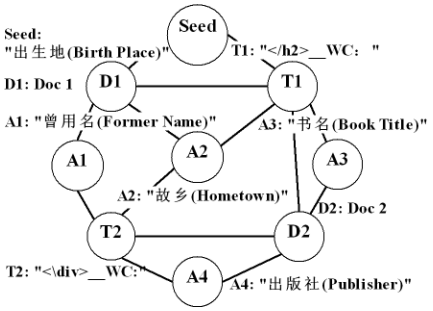

Figure 2: An example of our constructed graph.

work has addressed the problems to extract AV pairs form weak semi-text. The experimental results show that, when facing weak semi-text, our method outperforms the baseline method which does not leverage site-level knowledge.

2 System Description

The system consists of two part: the site-level extraction and the page-level extraction (Figure 1). Site-level extraction aims to obtain site-level knowledge from pages of a website. Page-level extraction leverages obtained site-level knowledge to help the AV pair extraction from each page.

2.1 Site-level Extraction

We describe details of the modules in site-level extraction (Figure 1).

2.1.1 Weak Semi-block Identification and Attribute Template Learner

We first segment a webpage into several blocks based on the paragraph HTML tags. Then we align the initial attributes to text of each block. The aligned attributes are used to induce templates to extract more attributes. A template is composed of a prefix and a separator. The separator is referred to the character or word next to the matched attribute and the prefix means characters previous to it. We take the string which begins at the head of first html tag before the matched attribute and ends at the head of the matched attribute as the template's prefix. For example a HTML fragment "⋯<*div class="spctrl"*></*div*> 性 别*(Sex)*：男*(Male)*⋯", in it, "性别*(Sex)*" is the attribute, "</*div*> " is the prefix and "： " is the separator, the template is " WC： " where WC is a placeholder for the attribute. And we set the prefix's window size as 15. If no html tag has been found within the window, then the template of this attribute is abandoned. Finally, we obtain a collection of templates of the weak semi-block.

We employ heuristic rules based on aligned attributes number and types and templates number to judge whether a block is a weak semi-block or not. Settings of two rules are discussed in the experiment at Section 5.4.1. They are: i) number of strings matched to initial attribute list is no smaller than $t1$, and ii) sum of attributes' probabilities having been matched to strings in the text is larger than $t2$. We represent them with a parameter vector $T = (t1, t2)$.

2.1.2 Attribute Extraction

The obtained templates are used to extract more attributes in each block. Intuitively, more frequent a template is found in a weak semi-block, more likely a string extracted by that template is an attribute. Based on this idea, templates with higher frequencies will have higher priority than those with the lower frequencies when extracting attributes. After we run through all the pages of the site, we get a collection of templates and attributes. Then we rank them to obtain site-level knowledge.

2.1.3 Ranker

To rank obtained templates and attributes to get site-level knowledge, we use the graph walk based technique (Wang and Cohen, 2007)(Wang and Cohen, 2009).

In the graph (Figure 2), attributes in initial attribute list are used as seeds. And these seeds are used to match the attributes in weak semi-block of a document (or a page) to learn templates. Then these templates are used to extract new attributes from the weak semi-block of a document (or a page). Intuitively, we consider that seeds appearing frequently are with high quality, templates derived by these seeds are tend to have good quality, and documents containing these seeds and templates are also deemed as high quality. Inversely, high quality documents also produce high quality attributes and high quality templates.

We utilize random walk with restart (RWR) to provide relevance score between two nodes (Tong et al., 2006). After the computation, we rank the attributes and templates by their probabilities in the final state vector.

We further refine the obtained ranked attributes by filtering obvious errors and the low ranks (site-level attributes) and generalize the top ranked templates by some rules (site-level templates). Site-level attributes and site-level templates composed the site-level knowledge.

2.2 Page-level Extraction

This section describes modules in page-level extraction (Figure 1).

2.2.1 Weak Semi-block Identification and AV Pair Extraction

To identify weak semi-block, we take the advantage of site-level knowledge to make several empirical rules based on the alignment of site-level templates and text of each block. The strings extracted by the templates are attribute candidates (*AttCandi* for short). We think only *AttCandies* extracted by authentic templates are correct attributes. A template is regarded as authentic once an *AttCandi* extracted by it exists in the site-level attributes. In the extraction of attribute's values, we follow the method in (Yoshinaga and Torisawa, 2007) with the hypothesis that an attribute immediately precedes its value, and another AV pair immediately follows those values.

2.2.2 Object Extraction

we need to obtain **objects** of AV pairs to form attribute relations (**OAV tuples**) mentioned in Section 1 (eg: *Kobe Bryant - DateOfBirth - August 23, 1978*). We inspect several sampled pages and find their shared unique HTML template of objects for AV pair in their own pages. And then use this shared template to extract objects in each pages.

3 EXPERIMENTS

3.1 Experiment Settings

We carry out the experiments on 3 million Baidu Baike[1] (Baike for short) pages. In them, 1/3 of the pages (observed from our sampling) contain weak semi-text. For pre-processing, we remove infoboxes in each page which are strict semi-text.

[1]http://baike.baidu.com/

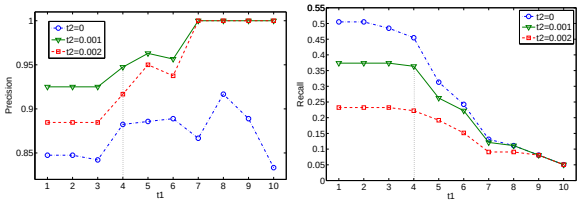

Figure 3: P/R curves with different $T = (t1, t2)$.

We evaluate on two aspects where site-level knowledge takes effect, they are: 1) weak semi-block identification in page-level extraction; 2) AV pair extraction in page-level extraction. We randomly sample 300 pages for manually labeling. 99 in them contain weak semi-blocks, and 1022 OAV tuples are labeled in the 99 pages. We use the manually labeled data as benchmark.

3.2 The Baseline

To demonstrate the effectiveness of incorporating the site-level knowledge, we implement a baseline system similar to Yoshinaga and Torisawa (Yoshinaga and Torisawa, 2007), which does not utilize the site-level knowledge. For comparison with our method, unlike their work which obtains initial attributes via search engine by manually generated regular expressions (it is hard to repeat precisely), we use the same initial attributes (attributes in infoboxes of Chinese Wikipedia) with our system as input.

3.3 Evaluation on Weak Semi-text

3.3.1 Evaluation on Weak Semi-block Identification (Ours vs. Baseline)

For weak semi-block Identification, we vary parameter vectors $T = (t1, t2)$ (Section 3.1) to show the selection of parameters. We set $t1 = \{x : 1 \leq x \leq 10\}$, $t2 = \{0, 0.001, 0.002\}$. Details of their effects to precision curves and recall curves are shown in Figure 3.

Since the contradiction between precision and recall in figure 3, we think high precision is more important comparing to high recall. For that, if we fail to recall a weak semi-block, we still have chance to get the same features this weak semi-block contains from others in the same site and recall it when doing page-level extraction with the help of site-level knowledge, however, if we identify the incorrect weak semi-block, the incorrect knowledge in it will be added to site-level knowledge which will bring amount of errors to our results when utilizing it to help page-level extrac-

Table 1: Performances of weak semi-block location.

	Output number	Correct	Precision	Recall	F-measure
$Baseline\_T_\beta$	12	12	**1.0**	0.121	0.216
$Baseline\_T_\gamma$	59	50	0.847	0.505	0.633
$Baseline\_T_\alpha$	38	36	0.947	0.364	0.526
$SiteExt\_T_\alpha$	100	96	0.96	**0.970**	**0.965**

Table 2: Strict and loose precision (P), recall (R) and F-measure (F) comparison of OAV tuple acquisition.

	P-strict	R-strict	F-strict	P-loose	R-loose	F-loose
$Baseline\_T_\beta$	0.822	0.159	0.266	0.888	0.171	0.287
$Baseline\_T_\gamma$	0.691	0.356	0.470	0.736	0.380	0.501
$Baseline\_T_\alpha$	**0.856**	0.307	0.452	**0.918**	0.330	0.485
$SiteExt\_T_\alpha$	0.844	**0.770**	**0.805**	0.887	**0.810**	**0.847**

tion. Therefore, we choose T as $T_\alpha = (4, 0.001)$, for our system (**SiteExt**), which gives a relatively higher recall with a high precision (Figure 3).

We compared $SiteExt\_T_\alpha$ with $T = T_\alpha$ and the baseline system which respectively uses T_α, $T_\beta = (7, 0.001)$ and $T_\gamma = (2, 0)$. The weak semi-block identification module of the baseline system is the same with the weak semi-block identification module of SiteExt in site-level extraction (Section 3.2). Therefore the results in these two modules are the same. From Figure 3, we can see that $Baseline\_T_\beta$ brings the highest recall within the ones bringing highest precision, and $Baseline\_T_\gamma$ brings the highest precision within the ones bringing highest recall.

Table 1 shows that $SiteExt\_T_\alpha$'s performance has a dramatic improvement comparing to other baseline systems which do not leverage site-level knowledge. The reason is that site-level knowledge captures attributes and templates specific to Baike. Meanwhile, weak semi-blocks in each page of the same site also share these features. As a result, we can identify more weak semi-blocks and reduce the incorrect ones with the same initial attribute set.

3.3.2 Evaluation on Object-Attribute-Value (OAV) tuples (SiteExt vs. Baseline)

We then evaluate the results of OAV tuple extraction. For different items in an OAV tuple, we select different similarity-computing methods. Because objects and attributes in an OAV tuple are always short phrases only with several words, we consider them as correct when their similarity meets a strict merit. On the other side, the value often contains descriptive contents which have more words. A small size of noises is acceptable. Therefore, besides the strict merit, we further select a loose merit. The two merits are shown in (3) and (4).

$$S_{loose} = \frac{len(wd(V_{bm} \bigcap wd(V_{ext}))}{min(len(wd(V_{bm}), len(wd(V_{ext}))))} \quad (1)$$

$$S_{strict} = \frac{len(wd(V_{bm} \bigcap wd(V_{ext}))}{max(len(wd(V_{bm}), len(wd(V_{ext}))))} \quad (2)$$

Where V_{bm} and V_{ext} separately denote the string of an attribute's value in benchmark and in our extraction results, $wd(V)$ is a set of different words in V, and $len(s)$ means sum of words in a set s. In the experiment, we set the thresholds both as 0.75. When all the similarity scores of three items (object, attribute, value) exceed the threshold, the extracted OAV tuple is regarded as correct.

Table 2 shows the performance of different systems. Comparing to $Baseline\_T_\alpha$, $SiteExt\_T_\alpha$ has great improvements in recall and has a slightly loss in precision. $SiteExt\_T_\alpha$ outperforms the other two baseline systems in both precision and recall. The experiment results prove that site-level knowledge is quite essential and effective to promise a good performance when extracting OAV tuples from weak semi-text of the same website. The two systems use the same initial attribute set as input, our method can identify more weak semi-blocks and extract more OAV tuples. It also proves that our method is less sensitive to the initial attribute set.

4 Conclusion

In this paper, we propose a novel approach that acquires site-level knowledge via a graph-based random walk model and leverages such knowledge to extract attribute relations from weak semi-text. Experimental results show that we can significantly improve the performance of identifying weak semi-text and OAV tuple extraction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61070106, No. 61272332 and No. 61202329), the National High Technology Development 863 Program of China (No. 2012AA011102), the National Basic Research Program of China (No. 2012CB316300) and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201201).

References

E. Alfonseca, M. Pasca, and E. Robledo-Arnuncio. 2010. Acquisition of instance attributes via labeled and related instances. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 58–65. ACM.

A. Arasu and H. Garcia-Molina. 2003. Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 337–348. ACM.

M.J. Cafarella, A. Halevy, D.Z. Wang, E. Wu, and Y. Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.

V. Crescenzi, G. Mecca, P. Merialdo, et al. 2001. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the international conference on very large data bases*, pages 109–118.

M. Pasca and B. Van Durme. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837.

M. Paşca, E. Alfonseca, E. Robledo-Arnuncio, R. Martin-Brualla, and K. Hall. 2010. The role of query sessions in extracting instance attributes from web search queries. *Advances in Information Retrieval*, pages 62–74.

J. Reisinger and M. Paşca. 2009. Latent variable models of concept-attribute attachment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 620–628. Association for Computational Linguistics.

H. Tong, C. Faloutsos, and J.Y. Pan. 2006. Fast random walk with restart and its applications.

P. Venetis, A. Halevy, J. Madhavan, M. Paşca, W. Shen, F. Wu, G. Miao, and C. Wu. 2011. Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment*, 4(9):528–538.

R.C. Wang and W.W. Cohen. 2007. Language-independent set expansion of named entities using the web. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 342–350. IEEE.

R.C. Wang and W.W. Cohen. 2009. Character-level analysis of semi-structured documents for set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1503–1512. Association for Computational Linguistics.

N. Yoshinaga and K. Torisawa. 2007. Open-domain attribute-value acquisition from semi-structured texts. In *Proceedings of the 6th International Semantic Web Conference (ISWC-07), Workshop on Text to Knowledge: The Lexicon/Ontology Interface (OntoLex-2007)*, pages 55–66.