Finding Specification Pages According to Attributes

Naoki Yoshinaga[†‡]
† Japan Society
for the Promotion of Science
6 Ichiban-cho, Chiyoda-ku, Tokyo,
102-8471, Japan
n-yoshi at jaist.ac.jp

Kentaro Torisawa[‡]
‡ Japan Advanced Institute
of Science and Technology
1-1, Asahidai, Nomi, Ishikawa,
923-1292, Japan
torisawa at jaist.ac.jp

ABSTRACT

This paper presents a method for finding a specification page on the web for a given object (*e.g.*, "Titanic") and its class label (*e.g.*, "film"). A specification page for an object is a web page which gives concise attribute-value information about the object (*e.g.*, "director"-"James Cameron" for "Titanic"). A simple unsupervised method using layout and symbolic decoration cues was applied to a large number of web pages to acquire the class attributes. We used these acquired attributes to select a representative specification page for a given object from the web pages retrieved by a normal search engine. Experimental results revealed that our method greatly outperformed the normal search engine in terms of specification retrieval.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*linguistic processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Experimentation

Keywords: specification finding, web search, attribute acquisition

1. INTRODUCTION

This paper proposes a method for finding a specification page for a given object and its class label. Here, a "specification page" for an object means a page that provides information on the object's comprehensive set of *attributes*, which label what we want to know about the object (*e.g.*, the "cast" of a film object), in visually distinguishable ways such as tables and lists (the left side of Figure 1). With a normal search engine, which usually gives higher ranks to authoritative sites such as shopping sites, news articles or blogs which rarely include attributes (the right side of Figure 1), we have to wade through numerous pages to extract pieces of information about the object. Our aim is to provide as the top result a single page that includes the description of several attributes of the given object, without explicitly specifying those attributes.

Our system finds a specification page for an object based on a knowledge base of the attributes of its *class*. We first construct a knowledge base of class attributes by using a simple, general unsupervised method that takes advantage of attribute behaviors on specification pages. We then filter out irrelevant words from putative candidates through author-aware statistics that we call *site frequency*. At runtime our system retrieves pages including the object with a normal search engine, and then finds a representative specification page using the class attributes.

WWW 2006, May 22–26, 2006, Edinburgh, Scotland.
ACM 1-59593-332-9/06/0005.

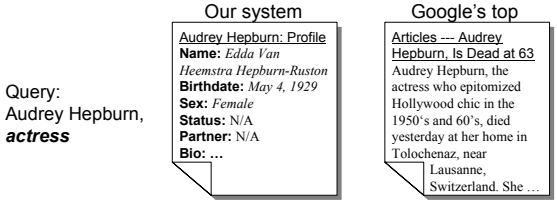

Figure 1: A Specification Page and Google's Top Result for Query "Audrey Hepburn, Actress"

2. RELATED WORK

This section reviews previous research on the automatic acquisition of attributes from texts. Two different types of cues have been used to identify object-attribute relations.

The first type is *lexico-syntactic patterns*, such as "the * of the x is," in which an object and its attribute often co-occur [1, 3]. Attributes acquired using lexico-syntactic patterns are less useful for finding specification pages because many attributes are referred to by different (synonymous) terms in normal texts and in specification pages.

The second type of cue, *layout information*, has been more widely studied as a direct way of describing attribute-value information on the web. Heuristics [2] and EM-based learning [5] have been proposed for finding attributes in HTML tables. The acquired attributes have been proven to be useful for collecting specification pages in a given *class* [4]. These methods, however, incur substantial computational costs in the table recognition process, and need carefully chosen data to acquire a comprehensive set of attributes.

Our method of acquiring attributes has a low computational cost and thus can handle a huge amount of data, so that a comprehensive set of attributes can be collected.

3. METHODOLOGY

This section describes our method of finding an informative specification page for a given object and its class label.

3.1 Construction of Attribute Knowledge Base

3.1.1 Sampling Candidate Pages from the Web

We first prepare the knowledge source for constructing the knowledge base of class attributes. We collect web pages that describe the target class as a topic by selecting web pages which have the target class label surrounded by certain HTML tags (`TITLE`, `H1`∼`H6`, `CAPTION`, `TD`, and `TH`). Only the text that follows the first appearance of the word in these tags is analyzed to acquire attributes.

3.1.2 Extracting Attributes from Candidate Pages

We acquire attributes from the candidate web pages in the following way. If a candidate page is a specification page, the

```
<H3>Titanic (year:1997)</H3> Director/ James Cameron <BR /> The
details:<TABLE><TR><TD>Starring</TD><TD>Leonardo DiCaprio, Kate
Winslet</TD></TR><TR><TD>Runtime</TD>194 min. </TD></TR></TABLE>
```

Figure 2: Example of Pattern Matching

attributes are likely to be emphasized in visually distinguishable ways through HTML tags and symbolic decorations, as illustrated in Figure 2. We thus collect expressions that are surrounded by the tags or braces, preceded by the prefixes, or followed by the suffixes (Figure 2), and that pass through a morphological analysis filter and a stop word filter.

3.1.3 Filtering Acquired Class Attributes

We next employ corpus-based statistics to filter out erroneously acquired attributes. The filter is used to rank the attribute candidates, and the top N (= 30 in our experiments) candidates are produced as attributes.

We adopted a novel metric called *site frequency*, *sf*, which worked better than *df-idf* and mutual information in the preliminary experiments. The site frequency $sf(x)$ for a candidate attribute, x, is defined as the number of *websites* from which x is acquired. Here, we treated a group of pages written/maintained by a single person/organization as a *website*, and defined a website for each page as a part of its URL, by digging through the URL until the directory included a file whose name matched `/^(?:index|default|main)\..+/`. The site frequency roughly expresses the number of authors who used the attribute to describe the class objects.

3.2 Finding Specification Pages

Given an input object and its class label, our system finds a representative specification page in two steps.

STEP 1: Extraction of page attributes: We collect candidate specification pages including the given object and the class label with a normal search engine, and extract attributes from each page using the method described in Section *3.1.2*.

STEP 2: Specification finding using the knowledge base: A representative specification page for an object x is selected from the candidate pages retrieved in STEP 1 by scoring each page p with attributes $\mathcal{A}_p$ according to the following function:

$$score(p) = \frac{\#(\mathcal{A}_p \cap \mathcal{A}_C) \times ratio(\mathcal{A}_p, \mathcal{A}_C)}{ave(\mathcal{A}_p, p) \times text\_size(x, p)},$$

where $\mathcal{A}_C$ is the class attributes. $\#(\mathcal{A}_p \cap \mathcal{A}_C)$ is the number of overlapping attributes between $\mathcal{A}_p$ and $\mathcal{A}_C$, reflecting the fact that good specification pages for x should include a large number of attributes of its class. $ratio(\mathcal{A}_p, \mathcal{A}_C)$ is defined as $\frac{\#(\mathcal{A}_p \cap \mathcal{A}_C)}{\#\mathcal{A}_p}$, which indicates that most of the attributes of x should be included in the class attributes. On the other hand, $ave(\mathcal{A}_p, p)$ is the average number of appearances of attributes, $a \in \mathcal{A}_p$, on the page p. This term is employed to favor a specification page only for the target object. $text\_size(x, p)$ is calculated as the length of the text surrounded by HTML tags that *first* contains the object label on the page. This term is used to select a page that is relevant to object x.

4. EXPERIMENTS

We constructed a 0.7 TB non-restricted web repository in Japanese, and then evaluated our system with a knowledge base of class attributes acquired from the repository.

Given a total of 100 objects for ten classes (suggested by our colleagues who were neither authors of this paper nor subjects in the experiments), three human subjects were asked to determine four attributes they wanted to associate with objects of each class without looking at the objects we had prepared. The subjects next examined whether each page produced by three systems (Google, SP, and SP⋆) referred to the target object in the given class, and then counted the number of attributes (or their synonyms) and their values that were included in the page. Google outputs the top results of the Google search engine. SP and SP⋆ select a specification page using our scoring function in Section 3.2; SP selects a page from the top 30 results provided by Google search engine, while SP⋆ selects a page from 10,000 pages randomly selected from the local web repository. A page was said to include an attribute-value pair only when a correspondence between the attribute and its value could be visually recognized as on the left side of Figure 1.

Table 1: Experimental Results (for objects where all systems output pages that referred to the objects)

Class Name	# objects	Google	SP	SP⋆
digital camera	4/10	1.50	**3.08**	**1.33**
racehorse	6/10	**4.00**	**4.00**	3.33
baseball player	1/10	0.33	**1.33**	0.33
actress/actor	4/10	**1.58**	1.50	1.00
hospital	4/10	0.33	**3.33**	1.25
corporation	0/10	NA	NA	NA
wine	5/10	1.53	1.53	**2.73**
museum	6/10	0.28	3.00	**3.33**
paperback	7/10	**3.33**	2.67	1.95
amusement park	5/10	1.07	2.80	**3.13**
weighted mean	42/100	1.81	**2.75**	2.33

Table 1 shows the performance of the three systems which together generated 126 pages for 42 objects[1]: all three systems provided pages that referred to the object. The columns titled Google, SP, and SP⋆ indicate the average number of attributes included in each page produced by each of these systems. SP and SP⋆ were superior to Google. This is interesting because SP⋆ used only 10,000 pages taken from less than 10% of Google's web repository and it did not use any page popularity criteria. This demonstrates that our attribute-driven approach is effective for our task. As SP achieved the best total performance of the three systems, a combination of page popularity criteria and our attribute-based scoring should yield an optimal system.

5. CONCLUSION

We have proposed a method that finds a specification page for a given object from the web. Our system found a specification page that included on average 2.75 (cf. 1.81 with Google) out of the four attributes the subjects had expected for each object. This result is promising since the current evaluation criteria is strict (some attribute labels, such as telephone number and manufacturer, are often omitted).

Although our current system needs a class label together with an object name, we plan to develop a module that will find an appropriate class label for a given object name.

6. REFERENCES

[1] A. Almuhareb and M. Poesio. Attribute-based and value-based clustering: An evaluation. In *Proc. EMNLP*, 2004.

[2] H.-H. Chen, S.-C. Tsai, and J.-H. Tsai. Mining tables from large scale html texts. In *Proc. COLING*, 2000.

[3] K. Tokunaga, J. Kazama, and K. Torisawa. Automatic discovery of attribute words from web documents. In *Proc. IJCNLP 2005*, pages 106–118, 2005.

[4] M. Yoshida and H. Nakagawa. Specification retrieval – how to find attribute-value information on the web. In *Proc. IJCNLP 2004*, pages 338–347, 2004.

[5] M. Yoshida, K. Torisawa, and J. Tsujii. A method to integrate tables of the World Wide Web. In *Proc. WDA*, 2001.

[1] Google, SP, and SP⋆ outputted a page that referred to the object for 81, 75, and 65 objects, respectively.