Open-Domain Attribute-Value Acquisition from Semi-Structured Texts

Naoki Yoshinaga[1] and Kentaro Torisawa[2]

[1] Japan Society for the Promotion Science,
6 Ichibancho, Chiyoda-ku, Tokyo, 102-8471, Japan,
n-yoshi@jaist.ac.jp
[2] Japan Advanced Institute of Science and Technology,
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan,
torisawa@jaist.ac.jp

Abstract. This paper proposes an unsupervised method that acquires a set of attribute-value pairs (AVPs, *e.g.*, ⟨director, W. Wyler⟩) for a given object (*e.g.*, "Ben-Hur") from semi-structured HTML documents. The objects' AVPs are one of the principal components of domain ontologies. We first acquire class attributes that are used by many web authors to describe the objects' AVPs. Then, we exploit the acquired class attributes to induce patterns for extracting AVPs from web pages. Experimental results show that, with our method, at least one set of correct AVPs are acquired for 67.7% of objects among open-domain class-object pairs whose source documents (web pages) include the objects' AVPs in layouts.

Key words: open-domain attribute-value acquisition, semi-structured texts, question answering, faceted search

1 Introduction

This paper proposes an unsupervised method that extracts a set of *attribute-value pairs* (AVPs, *e.g.*, ⟨director, W. Wyler⟩) for a given object (*e.g.*, "Ben-Hur") from the web. Here, attributes of an object are the words (*e.g.*, director) that refer to those aspects of the object in which people are interested (*e.g.*, "a person who directed a film object").

Objects' AVPs have been studied as one of the principal components of domain ontologies [1], and have been found beneficial in a wider range of NLP/IR applications such as question-answering (Q&A) [2] and faceted search [3]. The automatic extraction of object' AVPs has been attempted by several NLP researchers [4, 2, 5] to build a knowledge base for Q&A systems (*e.g.*, Q: "Tell me the *height* of Mt. Everest."— A: "*8,848m*"). Objects' AVPs can also provide *orthogonal categories* of the objects (*e.g.*, "films whose *director* is W. Wyler" — "Roman Holiday", "Ben-Hur", ...), and such categories can play the role of *facets* in faceted search (*e.g.*, the Flamenco system[3]), which receives much attention from IR researchers these days.

[3] http://flamenco.berkeley.edu/

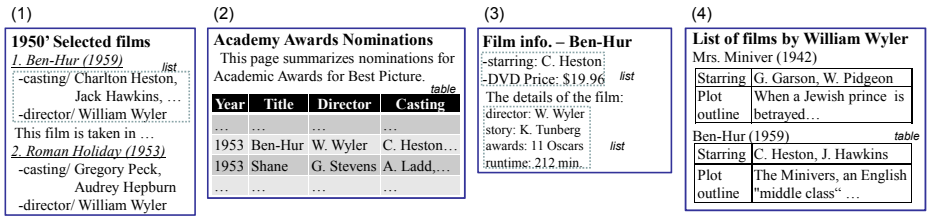

Fig. 1. Web pages that show *Ben-Hur*'s AVPs

The aim of this study is to acquire AVPs for a wide range of objects in an unsupervised manner. Our procedure finds specific layouts from the web (tables or lists, as shown in Fig. 1) that describe objects, and then extracts AVPs for the objects from these layouts. The key idea in our AVP acquisition is to use attributes of classes, to which the objects belong, as clues to induce regular expression patterns for extracting AVPs for objects from the layouts. We acquire class attributes which are used by many web authors to describe the class's objects in the layouts by using an author-aware scoring function called *site frequency*. Our method is applicable to several types of web layouts, as long as the layouts include the acquired class attributes.

The rest of this paper is organized as follows. Section 2 describes our method of acquiring AVPs from web pages. Section 3 reports experimental results. Section 4 discusses previous approaches to AVP acquisition. Section 5 concludes this paper and mentions future directions of this study.

2 Proposed method

In this section, we describe our method of acquiring AVPs for a given object (*e.g.*, "Ben-Hur"). Currently, we assume that the name of *a class* to which the given object belongs is also given (*e.g.*, "film"). The class name is needed to find *attributes* for the objects through the class name. In the future, we will automatically assign class names to input objects by using large-scale open-domain class-object databases [6].

Our AVP acquisition consists of two phases. First, we acquire attributes for a given class. Second, using the acquired class attributes, we find web pages that are likely to include the object's AVPs and then extract AVPs from these pages by using automatically induced regular expressions.

In what follows, we first describe a method for acquiring attributes for a given class, and then present a method for acquiring AVPs for objects in the class.

2.1 Attribute Knowledge Base Construction

We acquire class attributes by basically following the procedure proposed by Yoshinaga and Torisawa [7], which consists of the following three steps, namely:

- "X(の)Y" (Y of X), where $Y \in$ { リスト/一覧/名簿/目録 (list), 図鑑/案内/ガイド (guide), データ (data), 名鑑 (directory), データベース (database), カタログ (catalog), 情報 (info.), 評価 (rating), ランキング (ranking), 選び (choice), レポート (report), 解説 (note), アーカイブ (archive), 概要 (summary), 紹介 (intro.), 詳細 (detail), 感想 (impression), 特集 (focus)
- X について (About X)
- X (which appears as a suffix of the title string)

Fig. 2. Queries to collect pages from which we acquire attributes of class X (Japanese terms used in our experiments are followed by English translations)

```
<H3>Ben-Hur(year:'59)</H3>Director/ W. Wyler<BR />The details:<TABLE>
<TR><TD>Starring</TD><TD>C. Heston, J. Hawkins</TD></TR><TR><TD>Runtime
</TD><TD>212 min.</TD></TR></TABLE>
```

Fig. 3. Emphasized attributes in a web page

Table 1. HTML tags and text decorations used for identifying attributes

HTML tags: TD, TH, LI, DT, DD, B, STRONG, FONT, SMALL, EM, TT
Brace: 〔-〕, 【-】, 《-》, ［-］, 〈-〉, <->, [-] <->
Prefix: ∗, ✻, ●, ○, ■, □, ・, ◆, ◇, ★, ☆, ◎, ・, ○ , ◎
Suffix: : , :, ／, /, =

i) sampling pages that are likely to include class attributes, ii) extracting class attribute candidates and iii) filtering out erroneous attribute candidates. In the following, we describe each step in detail.

Sampling Web Pages from the Web: We first prepare web pages from which we extract class attributes. We collect web pages that show catalogs or specifications of the objects because these pages often contain attributes of the class objects, as shown in pages 3 and 4 in Fig. 1. In practice, we retrieved at most 2,000 pages that included strings referring to typical captions for the catalogs or specifications of the class objects (*e.g.*, "list of films", Fig. 2), on the positions of the page title (regions enclosed by `title` tags).

Extraction of Attribute Candidates: In the web pages retrieved in this section, the attributes are likely to be emphasized in visually distinguishable ways through HTML tags and symbolic decorations, as illustrated in Fig. 3. We employ such HTML tags and symbolic decorations as clues to acquire class attributes from the web pages. More precisely, we regard noun sequences that are surrounded by particular HTML tags or braces, preceded by prefixes or followed

C の A が (A of C [verb]), C の A に (at/in A of C), C の A より (from/than A of C)
C の A は (A of C [verb]), C の A で (by/with A of C), C の A から (from A of C)
C の A を ([verb] A of C), C の A へ (to A of C), C の A まで (even/until A of C)

Fig. 4. Pattern queries to eliminate irrelevant attributes A for class C

by suffixes, to be candidate attributes. We used patterns based on HTML tags and symbolic decorations shown in Table 1 to recognize candidate attributes.

Filtering Acquired Class Attributes: We use the statistics and hit counts of a commercial search engine to filter out erroneously acquired attributes, and then obtain a final set of attributes that includes attributes used by many web authors to describe objects in the class.

We then define a *site frequency* for each acquired attribute A, which directly reflects the number of authors who used that attribute to describe objects in the class, and rank the attributes according to site frequency to use the top n attributes in the following AVP acquisition:

Definition 1 (Site Frequency (SF)). *The* SF *for an attribute A extracted from a set of web pages D is the number of distinct authors of pages $d \in D$.*

In this study, we consider that web pages whose URLs share the same host name and directories are described by a single author. We then calculate the SF of A as the number of variations in the host name + directories of URLs of web pages from which the attribute A is extracted.

We further filter out erroneously acquired attributes from the ranked list of the candidate attributes. For each candidate attribute A for the class C, we obtain hit counts for queries that express lexico-syntactic patterns for class-attribute relation such as "C の A (A of C)". We employed the patterns shown in Fig. 4, which are used by Tokunaga et al. [8]. When the sum of hit counts for all the queries is 0, we remove the attribute from the candidate attributes.

2.2 Automatic AVP acquisition from the Web

Using class attributes acquired with the method described in Section 2.1, we extract AVPs for a given object according to the steps shown in Fig. 5. The basic idea is to induce regular expressions for extracting AVPs from layouts by using the contexts in which the class attributes appear.

Our AVP acquisition consists of the following steps. we first retrieve web pages that are likely to include AVPs for a given object (Step 1). We next enumerate regions that include objects' AVPs (AVP *blocks*) from each page retrieved in Step 1 (Step 2). We then recognize symbolic patterns for emphasizing attributes in each block, and then induce patterns to extract each AVP from that block (Step 3). We finally apply the pattern induced in Step 3 to each block and acquire AVPs from it (Step 4). In what follows, we describe each step in detail.

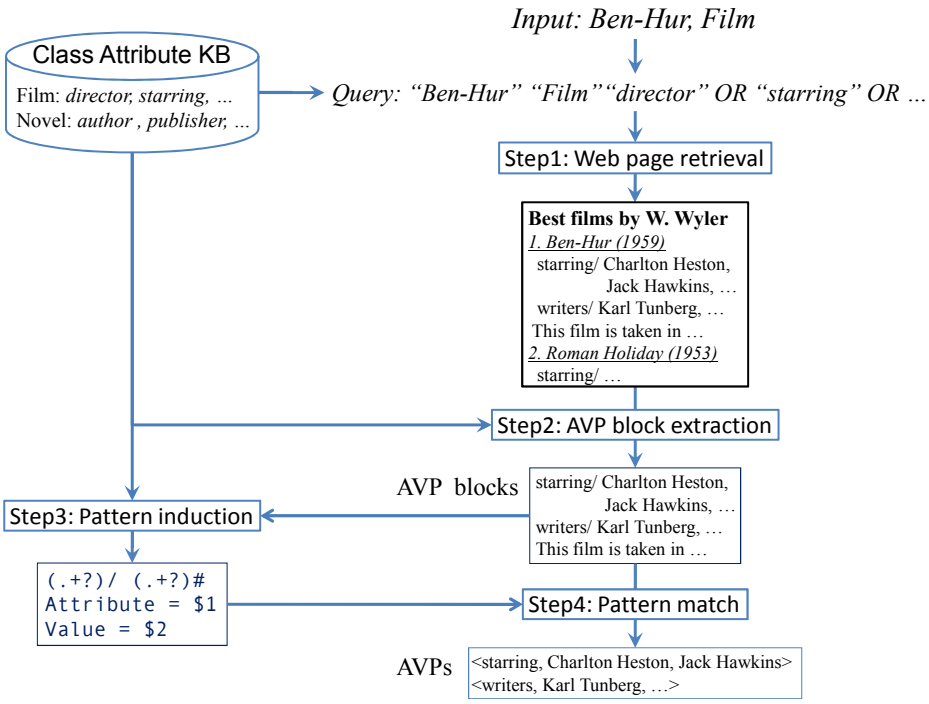

Fig. 5. Attribute-driven AVP Acquisition

Step 1. Collecting pages that include AVPs: To collect web pages that are likely to include AVPs for the input object, we use a commercial search engine with a query consisting of the object, the class, and the class attributes combined with 'OR' operators (see Fig. 5).

Step 2. Identifying AVP blocks: In Step 2, we identify tables or lists that describe the object's AVPs in the web pages retrieved in Step 1. The pages may even include layouts that describe AVPs for other objects (*e.g.*, those for "Roman Holiday" in page 1 and "Mrs. Miniver" in page 4 of Fig. 1). We search where objects and their AVPs appear in web pages, and hypothesize the positional relations between objects and layouts that describe the AVPs for them as follows:

Hypothesis I: AVPs for a given object are likely to appear concentrated in a small number of regions (AVP *blocks*) enclosed by HTML block tags. An AVP block describing an object should include attribute words, and also include or closely follow the object name.

We process each occurrence of the object name in the page from first to last, and iteratively collect innermost blocks enclosed by block tags such that the blocks include the input attributes preceded by the appearance of the object name as

candidate AVP blocks. For example, on page 3 of Fig. 1, when we have "Ben-Hur" as an input object and "director" and "starring" as the class attributes, we obtain two blocks that describe AVPs: a list with "starring" and "DVD price", and a list with "director", "story", "awards", and "runtime".

Step 3. Inducing AVP Extraction Patterns: We then induce patterns for extracting AVPs from each AVP block collected in Step 2. To design the induction procedure, we made the following assumption:

Hypothesis II: In an AVP block, an attribute immediately precedes its value, and another AVP immediately follows those values. When attributes in an AVP block are emphasized by some HTML tags, braces, prefixes or suffixes, the same symbols are used to emphasize other attributes in that block.

Following this hypothesis, our procedure induces a regular expression as a generic pattern for extracting attributes and their values. The procedure first applies to the AVP block the same patterns that we used for extracting attributes (in Section 2.1), and utilizes the matched strings that include the class attributes to identify HTML tags or symbolic cues that are used to emphasize attributes in the block. For example, when we have 'starring' and 'runtime' as class attributes and process an AVP block of the first listing in page 3 of Fig. 1, we obtain the matched string '-starring:', which contains the class attribute surrounded by the symbolic decorations '-' and ':'. By replacing the class attribute in this string with a wildcard that matches the shortest string, we acquire a regular expression `/-(.+?):/` for extracting attributes in this block. A value for an attribute is a string that spans from immediately after the corresponding attribute to just before another attribute. By putting the special character '`#`' before the attributes in the block, we thus obtain a generic pattern for extracting attributes and values in this example as `/-(.+?):(.+?)#/`.

The induced pattern can extract AVPs for attributes other than the class attributes the procedure used (*e.g.*, "DVD price", "director", "story", and "awards" in page 3 of Fig. 1). We can thereby extract all the AVPs in the block even when we have only one class attribute included in the AVP block.

Step 4. Acquiring AVPs from AVP blocks: We iteratively acquire AVPs from each AVP block by applying to the block the extraction pattern induced for the block in Step 3. We merge all the AVPs acquired from the blocks and return the resulting AVPs. When the pattern extracts the same attributes twice, we stop the acquisition and return the acquired AVPs until that point. This is done because, when the same attribute appears twice in the page, it usually expresses an AVP for different objects.

The above process (also Hypothesis II) assumed that the AVP block is described by a list. When we fail to recognize AVPs from AVP blocks enclosed by `th` or `td` tags (which express a single cell in a table), we attempt to recognize AVPs from the whole table structure. We can extract AVPs from those tables by regarding a row or a column that contains the maximum number of class attributes as expressing attributes. We omit further details due to space limitations.

3 Experiments

In this section, we report some experimental results of our AVP acquisition. We first prepared open-domain class-object pairs in Japanese, and then acquired AVPs for these objects with our method.

3.1 Experimental Settings

We obtained a development/test set of class-object pairs from hyponymy relations automatically acquired from the definition sentences in Wikipedia [6][4]. These hyponymy relations included not only class-subclass relations but also various class-instance relations. We sorted hypernyms in the hyponymy relations according to the number of hyponyms, and selected the top 1,000 hypernyms to obtain major hypernyms with a large number of hyponyms. We manually eliminated hypernyms from those hypernyms when they were too abstract (*e.g.*, "形式 (style)", "銘柄 (brand)"). We also replaced hypernyms with more general hypernyms when they were too specific (*e.g.,* "フランスの作曲家 (French composer)" → "作曲家 (composer)") or when they referred to a role/relation associated with other objects ("関連会社 (affiliate company)" → "会社 (company)"). The number of resulting hypernyms was 620. Next, we picked up objects from the hyponyms of the 620 hypernyms. Since the resulting hyponyms included not only *instances* but also concepts which do not have specific values for their attributes, we automatically eliminated from those hyponyms those which cannot be instances. More precisely, we eliminated hyponyms when they also appeared as hypernyms in the original hyponymy relations, or when the hit count of a query "どの X (which X)" was greater than 0 since generic concepts appear in this pattern (*e.g.*, "which car") [9]. We thereby obtained open-domain class-object pairs.

3.2 Experimental Results

We then acquired attributes for the 620 classes in the above class-objects pairs. We used Yahoo! API[5] to collect web pages for acquiring class attributes (Section 2.1). We randomly split the 620 classes into two sets with the same size, and used one as the development set and the other as the test set. Since the main aim of the following experiments was to show the utility of the class attributes in the AVP acquisition task, we analyzed the performance of AVP acquisition for classes for which reliable class attributes were available. We observed that the acquired attributes with a larger site frequency (SF) were more reliable than those with a smaller SF. To obtain classes that are likely to have at least one reliable class attribute, we then sorted the test classes according to the value of the largest SF among those for the attributes acquired for each class, and then used the top 20 classes for evaluation (left column in Table 2). We note that the resulting classes included various classes that had few common attributes. Five

[4] http://ja.wikipedia.org/
[5] http://developer.yahoo.co.jp/

Table 2. Top six class attributes acquired from the Web

CLASS	ACQUIRED ATTRIBUTES
武器 (weapon)	攻撃力 (offensive power), 名称 (name), 武器名 (weapon name), 備考 (remarks), 価格 (price), ブーメラン (boomerang)
会社 (company)	資本金 (capital fund), 事業内容 (description of business), 所在地 (location), 代表者 (representative), 従業員数 (# employee), 設立 (foundation)
機関 (agency/organization)	電話番号 (phone number), 所在地 (location), 電話 (phone), 住所 (address), 医療機関名 (name of medical institution), 診療科目 (department)
ホテル (hotel)	住所 (address), ホテル名 (name of hotel), チェックイン (check-in), 駐車場 (parking), チェックアウト (check-out), 料金 (rate)
選手 (player)	生年月日 (birthdate), 背番号 (uniform number), ポジション (position), 身長 (height), GK (goal keeper), 監督 (manager)
公園 (park)	所在地 (location), 駐車場 (parking), 電話 (phone), 住所 (address), 交通 (transportation), 場所 (place)
車両 (wheeled vehicle)	年式 (model year), メーカー (manufacturer), グレード (grade), 走行距離 (mileage), 排気量 (engine displacement), 仕様 (specification)
行事 (event)	場所 (place), 日時 (date), 内容 (content), 会場 (venue), 講師 (lecturer), 参加費 (entrance fee)
営業所 (sales office)	住所 (address), 営業時間 (office hour), 本社 (head office), 電話番号 (phone number), 所在地 (location), 電話 (phone)
規則 (rule)	施行期日 (effective date), 趣旨 (point), 目的 (purpose), 経過措置 (interim measure), 定義 (definition), 委任 (commission)
医者 (medical doctor)	院長 (director), 副院長 (assistant director), 整形外科 (orthopedic surgery), 内科 (internal medicine), 外科 (surgery), 専門分野 (expertise)
食品 (foodstuff)	価格 (price), 内容量 (volume), 日時 (date), 商品名 (product name), ダイエット (diet), 原材料 (material)
制度 (institution)	電話 (phone), 問い合わせ先 (contact info.), 電話番号 (phone), 対象者 (object person), 参考 (reference), 出版社 (publisher)
ドラマ (TV drama)	出演 (casting), 脚本 (story), 発売日 (release date), 価格 (price), 原作 (original), 日時 (date)
会議 (meeting)	日時 (date), 場所 (location), 電話 (phone), 事務局 (organizer), 出席者 (participants), 会場 (venue)
ソフトウェア (software)	発売日 (release date), 価格 (price), 日時 (date), 販売価格 (sales price), メーカー (manufacturer), 商品名 (product name)
アニメ (animated movie)	監督 (director), 日時 (date), カテゴリー (category), 原作 (original), 価格 (price), 音楽 (music)
思想 (ideology)	カテゴリー (category), メーカー (maker), 著者 (author), 発売日 (release date), 目次 (content), 出版社 (publisher)
活動 (activity)	日時 (date), 場所 (place), 電話 (phone), 活動内容 (activity content), 内容 (content), 会場 (venue)
アルバム (album)	発売日 (release date), 価格 (price), 作詞 (lyric), CD (CD), 初回限定盤 (limited edition), 作曲 (composition)

objects were randomly chosen from the objects in the class-object pairs, and we obtained 100 test class-object pairs.

As we have described in Section 2.1, we use the top n attributes ranked by the SF, and we set $n = 6$ here. Table 2 shows the top six attributes used for acquiring AVPs. We obtained meaningful attributes for 18 classes other than "medical doctor" and "ideology". We failed to acquire correct attributes for "medical doctor" because objects in this class often appear together with objects in other classes ("hospital"). We failed to acquire correct attributes for "ideology" because "思想 (ideology)" often appeared in the substrings of names of objects present in other classes (*e.g.*, 'book').

We applied our method to the 100 class-object pairs and acquired AVPs for them. In Step 1, we retrieved the top ten pages by using Yahoo! API. Due to a

Table 3. Number of pages where the objects' AVPs were acquired

	correct AVPs	incorrect AVPs	AVPs not acquired	total
N° of pages w/ AVPs in layouts	97	22	90	209
N° of pages w/o AVPs in layouts	0	21	483	504
total	97	43	573	713

download failure, we processed on average 7.13 pages for each pair after Step 2 (713 pages in all).[6] We successfully acquired AVPs for 140 pages.

A human subject was then asked to evaluated our AVP acquisition. The subject first determined whether each of the 713 pages included the layouts describing AVPs for the given objects, and the subject judged that 209 pages had layouts that described the objects' AVPs.[7] The subject next analyzed these layouts to identify attributes, values, and their correspondences, and then checked whether the AVPs acquired from the page are the same as those he recognized.

This check allowed us to evaluate our AVP acquisition by using the following two criteria: 1) we first count the number of pages from which we acquired correct AVPs to show the accuracy/coverage of our method to acquire AVPs from each page, and 2) then count the number of class-object pairs for which we could acquire correct AVPs from at least one page to address the coverage of our method for (open-domain) objects.

Coverage/Accuracy of our AVP acquisition from web pages: Table 3 shows the number of pages for which we acquired correct AVPs for the given objects. Our method extracted correct AVPs from 97 web pages in terms of AVP acquisition from the pages, the coverage is 46.4% (97/209) and the accuracy is 69.3% (97/140). We acquired in total 683 AVPs from these 97 web pages (on average 7.04 AVPs per page). Table 4 shows the detailed analysis for all the AVPs extracted from the 140 pages. In the table, the column CORR shows the number of pages for which all the acquired AVPs are correct (the same as the ones the subject identified), while the column CORR* shows the number of pages for which all the acquired attributes are correct, but some of their values include *irrelevant strings* together with the correct values (*e.g.*, a string "William Wyler *This film is taken in ...*" as a value for the director of "Ben-Hur" in page 1 of Fig. 1). In this analysis, we consider (a set of) AVPs judged as CORR or CORR* as correct AVPs when we obtain the above-mentioned coverage and accuracy. The column ATTR INCORR shows the number of pages for which the method extracted AVPs from appropriate layouts, but the attributes are recognized incorrectly, while the

[6] We excluded web pages of Wikipedia since objects in the test set were obtained from Wikipedia entries and we will easily obtain AVPs from those pages.

[7] We asked the subject to consider that there were layouts for AVPs when at least one attribute-value relation for the objects could be identified by any visually distinguishable ways other than sentences. Although most cases were covered by our cues in Table 1, a few exceptions included tables expressed using spaces in plain texts.

Table 4. Classification of pages where the objects' AVPs were acquired

CLASS	CORR	CORR*	ATTR INCORR	OBJ INCORR	total
武器 (weapon)	4	1	5	1	11
企業 (company)*	14	0	2	0	16
機関 (organization/agency)	1	0	0	2	3
ホテル (hotel)*	8	2	1	0	11
選手 (player)*	3	0	0	0	3
公園 (park)*	5	0	0	3	8
車両 (wheeled vehicle)	0	0	0	1	1
行事 (event)	2	0	0	3	5
営業所 (sales office)	1	0	2	0	3
規則 (rule)	1	0	0	1	2
医師 (medical doctor)	1	0	0	3	4
食品 (foodstuff)*	13	1	1	0	15
制度 (institution)	1	0	2	1	4
ドラマ (TV drama)*	5	3	0	2	10
会議 (meeting)	1	0	1	1	3
ソフトウェア (software)*	2	1	0	2	5
アニメ (animation movie)*	10	1	2	2	15
思想 (ideology)	3	0	0	3	6
活動 (activity)*	1	0	0	1	2
アルバム (album)*	6	6	0	1	13
total	82(58.6%)	15(10.7%)	16(11.4%)	27(19.3%)	140
best-10 (*)	67(68.4%)	14(14.3%)	6(6.1%)	11(11.2%)	98

column OBJ INCORR shows the number of pages from which we extracted AVPs for irrelevant objects. When we focus on the best 10 classes (marked $*$ in the table), the accuracy reaches 82.7% (81/98).

Coverage of our AVP acquisition for open-domain objects: We then determined the number of class-object pairs for which correct AVPs are acquired. For the five objects in the 20 classes, the subject judged that, for 65 objects, there is at least one page that includes the objects' AVPs. In order to show the coverage of our method in terms of AVP acquisition for the objects, we calculated the number of pairs for which correct AVPs are acquired from at least one page. From the web pages for these 65 pairs, we could acquire at least one correct set of AVPs for 44 objects (67.7%). We also determined the classes for which we could acquire the objects' correct AVPs. We acquired correct AVPs for at least one object in 19 classes (95%), namely, all the classes other than "wheeled vehicle". We note that, when we extracted AVPs from more than one page for one pair, the probability that we acquired the correct AVPs drastically increases. For 35 class-object pairs for which we extracted AVPs from more than one page, we acquired correct AVPs for 33 pairs (94.2%). These results show that we can acquire AVPs with high accuracy/coverage by selecting the target classes appropriately.

4 Related Work

In this section, we first review existing approaches to acquire attributes of classes, and then mention other work to obtain AVPs from (web) documents.

4.1 Automatic attribute acquisition

Tokunaga et al. [8] acquired attributes for a given class by applying lexico-syntactic patterns to normal texts. Paşca and Durme [10, 11] have recently proposed a method that extracts class attributes from massive query logs by using seed objects and seed attributes. However, it is unclear whether attributes extracted from normal texts or query logs are useful for extracting AVPs in layouts. We have found some cases where, to refer to the same attributes, simpler terms are preferred, at least in normal texts, while more technical terms are used in layouts (*e.g.* to refer to "outside dimension" of "digital camera", "サイズ (size)" is preferred in normal texts, while "外形寸法 (dimension)" is often used in layouts). It is also difficult for people who are not employed by a commercial search engine vendor to obtain a large set of query logs, and it is desirable to have a method to acquire AVPs from freely available resources such as web pages.

4.2 Automatic AVP acquisition

Brin [12], Etzioni et al. [4], and Paşca et al. [13] proposed semi-supervised methods that used seed data to obtain patterns for acquiring AVPs from texts. Sekine et al. [5] used supervised learning to generate patterns for finding values of manually tailored attributes from texts. Kushmerick [14] advocated a supervised *wrapper induction* for general information extraction from semi-structured texts such as HTML documents, and AVP acquisition can be seen as a special case of this task. Although these (semi-)supervised approaches have been used to collect AVPs in a pre-defined domain, AVPs acquired with our method will help those methods to acquire AVPs for open-domain objects.

Takahashi [2] acquired objects' AVPs using lexico-syntactic patterns. However, it is extremely difficult to design such patterns for some value types such as sentences (*e.g.*, values of "plot outline" for "film"). Another problem is anaphora or omission in texts. Objects are often anaphorically referred to or often omitted in sentences (*e.g.,* "I watched Ben-Hur on TV. The director was W. Wyler"). Chen et al. [15] and Yoshida et al. [16] proposed unsupervised methods for finding AVPs from HTML tables in a certain domain. However, because tables do not always include the object names together with the AVPs (*e.g.*, "Ben-Hur" in page 4 in Fig. 1), we need to find the objects in the pages for which the tables describe AVPs. These studies did not address this problem, while we solved it by using the class attributes and heuristics.

5 Conclusion

This paper proposed an unsupervised method that acquires attribute-value pairs (AVPs) for a wide range of objects from layouts on the web. We exploit class

attributes acquired from the web, in order to induce patterns for extracting AVPs from the layouts. For each set of five objects in the 20 open-domain classes, we successfully extracted at least one set of correct AVPs for 44, or 67.7%, of the 65 objects whose source documents (web pages) included the objects' AVPs. Our method has been proven to be useful for acquiring AVPs for certain classes.

We will extend our method to handle other languages including English. We plan to apply our method to large-scale class-object relations [6], in order to construct a large-scale AVP database. We will exploit that database to support Q&A systems and to build a faceted search system for a wide range of objects.

References

1. Lenat, D.B., Guha, R.: Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Addison-Wesley (1990)
2. Takahashi, T.: Computation of Semantic Equivalence for Question Answering. PhD thesis, Nara Institute of Science and Technology, Nara, Japan (2005)
3. Broder, A., Maarek, Y., eds.: Proc. of the SIGIR Workshop on Faceted Search. Springer (2006)
4. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in KnowItAll (preliminary results). In: Proc. of WWW. (2004) 100–110
5. Sekine, S., Sudo, K., Ando, M.: Encyclopedia QA system using automatic discovery of attribute value and question sentence patterns. In: Proc. of the Annual Meeting of Natural Language Processing. (2005) (in Japanese).
6. Sumida, A., Torisawa, K.: Hacking Wikipedia for hyponymy relation acquisition. In: Proc. of IJCNLP. (2008) to appear.
7. Yoshinaga, N., Torisawa, K.: Finding specification pages according to attributes. In: Proc. of WWW. (2006) 1021–1022
8. Tokunaga, K., Kazama, J., Torisawa, K.: Automatic discovery of attribute words from web documents. In: Proc. of IJCNLP. (2005) 106–118
9. Sumida, A., Shinzato, K., Torisawa, K.: Concept-instance relation extraction from simple noun sequences using a search engine on a web repository. In: Proc. of the ISWC workshop on Web Content Mining with Human Language Technologies. (2006)
10. Paşca, M., Durme, B.V.: What you seek is what you get: Extraction of class attributes from query logs. In: Proc. of IJCAI. (2006) 2832–2837
11. Paşca, M.: Organizing and searching the World Wide Web of facts - step two: Harnessing the wisdom of the crowds. In: Proc. of WWW. (2007)
12. Brin, S.: Extracting patterns and relations from the World Wide Web. In: Proc. of the WebDB workshop at EDBT. (1998)
13. Paşca, M., Lin, D., Bigham, J., Lifchit, A.: Organizing and searching the World Wide Web of facts - step one: the one-million fact extraction challenge. In: Proc. of AAAI. (2006) 1400–1405
14. Kushmerick, N.: Wrapper induction: Efficiency and expressiveness. Artificial Intelligence **118**(1–2) (2000) 15–68
15. Chen, H.H., Tsai, S.C., Tsai, J.H.: Mining tables from large scale HTML texts. In: Proc. of COLING. (2000)
16. Yoshida, M., Torisawa, K., Tsujii, J.: A method to integrate tables of the World Wide Web. In: Proc. of WDA. (2001)