Unsupervised Extraction of Attributes and Their Values from Product Description

Keiji Shinzato **Satoshi Sekine**
Rakuten Institute of Technology
{keiji.shinzato, satoshi.b.sekine}@mail.rakuten.com

Abstract

This paper describes an unsupervised method for extracting product attributes and their values from an e-commerce product page. Previously, distant supervision has been applied for this task, but it is not applicable in domains where no reliable knowledge base (KB) is available. Instead, the proposed method automatically creates a KB from tables and itemizations embedded in the product's pages. This KB is applied to annotate the pages automatically and the annotated corpus is used to train a model for the extraction. Because of the incompleteness of the KB, the annotated corpus is not as accurate as a manually annotated one. Our method tries to filter out sentences that are likely to include problematic annotations based on statistical measures and morpheme patterns induced from the entries in the KB. The experimental results show that the performance of our method achieves an average F score of approximately 58.2 points and that filters can improve the performance.

1 Introduction

E-commerce enables consumers to purchase a large number of products from a variety of categories such as books, electronics, clothing and foods. In recent years, this market has grown rapidly around the world. Online shopping is regarded as a very important part of our daily lives.

Structured product data are most important for online shopping malls. In particular, product attributes and their values (PAVs) are crucial for many applications such as faceted navigation and recommendation. However, since structured information is not always provided by the merchants, it is important to build technologies to create this structured information (such as PAVs) from unstructured data (such as a product description). Because of the wide variety of product types, such technology should not rely on a manually annotated corpus. One of the promising methods for information extraction (IE) without a manually annotated corpus is *distant supervision* (Mintz et al., 2009), which leverages an existing knowledge base (KB) such as Wikipedia or Freebase to annotate texts using the KB instances. These popular KBs, however, are not very helpful for distant supervision in an e-commerce domain for the following reasons. (1) An infobox in a Wikipedia article is not always tailored towards e-commerce. For instance, as of May 2013, the infobox attributes of wine in English Wikipedia included energy, carbohydrates, fat, protein and alcohol[1]. These are not particularly useful for users seeking their favorite wines through online shopping. Instead, the grape variety, production area, and vintage of the wine would be of greater interest. (2) On the other hand, Freebase contains PAVs for limited types of products such as digital cameras[2]. However, since Freebase is currently only available in English, we cannot use Freebase in a distant supervision method for other languages. Moreover, the number of categories whose PAVs are available in Freebase is limited even in English.

In this paper, we propose a technique to extract PAVs using an automatically induced KB. For the induction, the method uses structured data such as tables and itemizations embedded in the unstructured data. An annotated corpus is then automatically constructed from the KB and unstructured data, i.e. product pages. Since these texts are in HTML format, we can extract the attribute candidates and their values using pattern matching with tags and symbols. We can expect the KB to

[1] http://en.wikipedia.org/wiki/Wine
[2] http://www.freebase.com/view/digicams/digital_camera

International Joint Conference on Natural Language Processing, pages 1339–1347,
Nagoya, Japan, 14-18 October 2013.

have a certain level of accuracy because the tables and itemizations are created by merchants who are product experts. However, there may be a need to group synonymous attribute names. We propose a method for grouping synonymous attribute names by observing the commonality of the attribute values and their co-occurrence statistics. The model for extracting attribute values is then trained using the annotated corpus to find PAVs of the products. Because the KB is incomplete, the annotated corpus may contain annotation mistakes; *false-positive* and *false-negative*. Thus, our method tries to filter out sentences that are likely to include those problematic annotations based on statistical measures and morpheme patterns induced from the entries in the KB.

The contributions of our work are as follows:

1. Unsupervised and scalable methods for inducing a KB consisting of PAVs, and for discovering attribute synonyms.

2. Unsupervised method for improving the quality of an automatically annotated corpus by discarding false-positive and false-negative annotations. These problematic annotations are always included in automatically annotated corpora although such corpora can be constructed without requiring expensive human effort.

3. Comprehensive evaluation of each component: Automatically induced KBs, annotation methodology, and the performance of IE models based on the automatically constructed corpora. In particular, the annotation methodology and the IE models are evaluated by using a dataset comprising 1,776 manually annotated product pages gathered from eight product categories.

To summarize, as far as we know, this is a first work to extract product attributes and their values relying solely on product data, and completing all the steps of this task, including KB induction, in a purely unsupervised manner.

2 Related Work

2.1 Product Information Extraction

Bing et al. (2012) proposed an unsupervised methodology for extracting product attribute-values from product pages. Their method first generates word classes from product review data related to a category using Latent Dirichlet Allocation (LDA) (Blei et al., 2003). The method then automatically constructs training data by matching words in classes and those in product pages for the category. After that, extraction models are built using the training data. Their method does not deal with attribute synonymy and problematic annotations in their training data. They evaluated the result of their extraction model only, while this paper reports on evaluation results of not only extraction models but also induced KBs and annotation methodology. In addition, their method employs LDA for generating word classes. This may involve the issue of *scalability* when running the method on large size real-world data. On the other hand, we employ a simple rule-based approach to induce the KBs. We can then straightforwardly apply the method on the large-scale data.

Mauge et al. (2012) also proposed methods to extract product attribute-values using hand-crafted patterns, and to group synonymous attributes in a supervised manner. They, however, only evaluated a part of the extracted attribute names, and aggregated synonymous attribute names. They did not evaluate the extracted attribute values.

Our work is also similar to Ghani et al. (2006), who construct an annotated corpus using a *manually tailored* KB and then train models using the corpus to extract attribute values. Probst et al. (2007) and Putthividhya and Hu (2011) also proposed a similar approach with the work of Ghani. The main difference between these works and ours is that our method does not require manually tailored KBs. Instead, our method automatically induces a KB of PAVs from structured data embedded in product pages.

In addition to the above, many wrapper based approaches have been proposed (Wong et al., 2008; Dalvi et al., 2009; Gulhane et al., 2010; Ferrez et al., 2013). The goal of these approaches is to extract information from documents semi-structured by any mark up language such as HTML. On the other hand, our method aims at extracting (product) information from full texts although the method leverages semi-structured documents to induce KBs.

2.2 Knowledge Base Induction

There are many works for automatically inducing KBs using syntactic parsing results (Rooth et al., 1999; Pantel and Lin, 2002; Torisawa, 2001; Kazama and Torisawa, 2008), and semi-

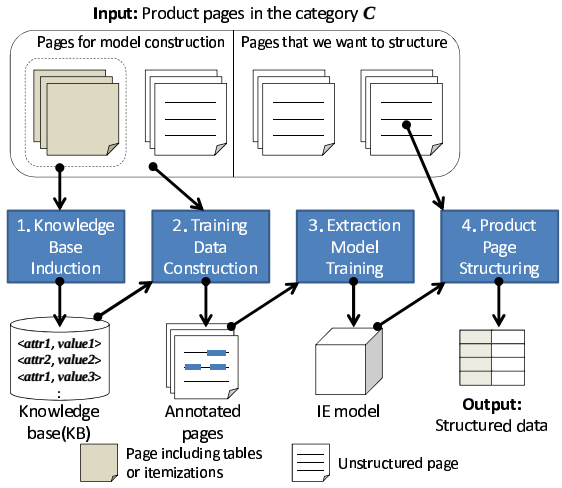

Figure 1: Overview of our approach.

structured data (Shinzato and Torisawa, 2004; Wong et al., 2008; Yoshinaga and Torisawa, 2006; Mauge et al., 2012). As clues for the induction of attribute-value KBs, previous works employ lexico-syntactic patterns such as *<attribute>:<value>* and *<value> of <attribute>*, and layout patterns in semi-structured data such as tables and itemizations marked up by HTML tags.

We employ similar clues as Yoshinaga and Torisawa (2006) for KB induction. In addition to the induction, our method tries to find synonymous attributes in the induced KB. This is the difference between Yoshinaga's work and ours. The method of Mauge et al. (2012) also finds attribute synonyms, but it requires supervision to find these.

3 Data

We used Rakuten's product pages comprising over 100 million products in 40,000 categories[3]. Each product is assigned to one category by merchants offering it on the Rakuten site. In contrast to Amazon, Rakuten's product pages are not well-structured, because the product pages are designed by the merchants. Although some pages include tables for describing product information, majority of pages describe product information in full texts. That's why we need to extract product information from the texts.

4 Approach

An overview of our approach is shown in Figure 1. The input for the approach is a set of product pages belonging to a single category like wine or shampoo. From the given pages, a value extraction model for the category is generated, and then the model is used to structure unstructured pages in the category.

A detailed explanation of steps 1, 2 and 3 in Figure 1 is given in the remaining sections. Explanation of step 4 is omitted because the extraction model is simply applied to a given page.

[3]http://www.rakuten.co.jp/

保存方法 (method of preservation), その他 (etc), 商品説明 (explanation), 広告文責 (responsibility for advertisement), 特徴 (characteristics), 仕様 (specs.)

Figure 2: List of stop words for attributes.

P1: `<T(H|D)>[ATTR]</T(H|D)><TD>[ANY]</TD>`
P2: `[P][ATTR][S][ANY][P]`
P3: `[P][ATTR][ANY][P]`
P4: `[ATTR][S][ANY][ATTR][S]`

Figure 3: Patterns for extracting attribute values. The [ATTR] tag denotes a collected attribute, the [ANY] tag denotes a string, the [P] tag denotes the prefix and *open*-braces, and the [S] tag denotes the suffix and *close*-braces. The prefix, suffix and brace are defined in (Yoshinaga and Torisawa, 2006). Attributes of HTML tags are removed before running **P1**.

4.1 Knowledge Base Induction

4.1.1 Extraction of Attribute-Values

A KB is induced from the tables and itemizations embedded in the given product pages. We first collected product attributes based on the assumption that *expressions that are often located in table headers can be considered as attributes*. The regular expression <TH.*?>.+?</TH> is run on the given pages, and expressions enclosed by the tags are collected as attributes. (The <TH> tag is used for a table header.) The collected candidate set includes expressions that can not be regarded as attributes. We excluded these using a small set of stop words given in Figure 2.

To extract values corresponding to the collected attributes, the regular expressions listed in Figure 3 are run on the given product pages. An expression that matches the position of [ANY] is extracted as a value of the attribute corresponding to [ATTR]. The first appearance of [ATTR] is selected as the attribute in **P4**. The extracted value and its attribute are stored in the KB along with the number of merchants that use these in the table and itemization data. Henceforth, we refer to this number as the *merchant frequency* (MF).

4.1.2 Attribute Synonym Discovery

The KBs constructed in the previous section still contains numerous synonyms; that is, attributes

with the same meaning, but different spellings, are included. This is because merchants do not have a standard method for describing products on their own pages. For example, "Bordeaux" and "Tuscany" can be regarded as production areas of wine. However, some merchants refer to "Bordeaux" as the *production area*, while others consider "Tuscany" to be the *region*. If we use KBs that include the incoherence as an annotation resource, corpora containing annotation incoherence are constructed. To avoid this problem, we set out to identify synonymous attributes in the KBs.

We attempt to discover attribute synonyms according to the assumption that *attributes can be seen as synonyms of one another if they have never been included in the same structured data, and they share an identical popular value*. We regard the MF of a value as a measure of *popularity*. Based on this assumption, for all combinations of attribute pairs with an identical attribute value whose MF exceeds N, we verify whether they appear simultaneously in structured data (i.e., table and itemization data). Attribute pairs satisfying the condition are regarded as synonyms. The value of N is defined by the equation $N = \max(2, M_S/100)$ where M_S is the number of merchants providing structured data in the category. As a result, we obtain a vector of attributes that can be regarded as synonyms, such as (*region*, *country*), and (*grape*, *grape variety*) for the wine category. We refer to this set of synonym vectors as S_{attr}.

Next, synonym vectors with high similarity are iteratively aggregated. For example, the vector (*region*, *country*, *location*) is generated from the vectors (*region*, *country*) and (*location*, *country*). The similarity between vectors v_a and v_b is computed by the cosine measure: $sim(v_a, v_b) = v_a \cdot v_b/|v_a||v_b|$. We iteratively aggregate the two vectors with the highest similarity in set S_{attr}. The aggregation process continues until the highest similarity is below the threshold th_{sim}, which we set to 0.5. The threshold value was determined empirically.

Table 1 shows an example of the KB for the wine category. We can see that the "variety" attribute includes three Japanese variants.

4.2 Training Data Construction

An annotated corpus for training a model for attribute value extraction is constructed from the given product pages and the KB built from those pages. Values in the KB are simply annotated for the pages. Then, sentences possibly including incorrect annotations or sentences where annotation are missing are automatically filtered out from the annotated pages to improve annotation quality.

Table 1: Example of the KB for the wine category. The top three attributes are listed with their top four values. ([*number*] denotes the MF of a value.)

Attribute	容量 (volume)	品種 (variety)	タイプ (type)
Attribute synonyms	内容量 (content)	ぶどう品種 (grape variety) ブドウ品種 (grape variety) 使用品種 (usage variety)	—
Attribute values	750ML[147] 720ML[64] 375ML[49] 500ML[41]	シャルドネ [59] (Chardonnay) メルロー [36] (Merlot) シラー [29] (Syrah) リースリング [29] (Riesling)	辛口 [34] (dry) 赤 [24] (red) 白 [23] (white) フルボディ[23] (full body)
# values	159	1,578	153

4.2.1 Attribute Value Annotation

All given product pages are split into sentences following block-type HTML tags, punctuation, and brackets. Each sentence is then tokenized by a morphological analyzer[4]. The longest attribute value matching a sub-sequence of the token sequence is annotated. We employed the Start/End tag model (Sekine et al., 1998) as chunk tags for the matched sequence. If the matched value corresponds to more than one attribute, the entry with the largest MF is selected for annotation. Note that if other attribute values are contained in the matched sub-sequence, they are not annotated.

4.2.2 Incorrect Annotation Filtering

Some attribute values with *low* MFs are likely to be *incorrect*. The quality of the corpus, and the performance of extraction models based on the corpus deteriorate if such values are frequently annotated. We detect incorrect value annotation in the corpus according to the assumption that *attribute values with low MFs in structured data (i.e., tables and itemizations) and high MFs in unstructured data (i.e., product descriptions) are likely to be incorrect*. Thus, we designed the following score:

$$Score(v) = \frac{MF_D(v)/N_M}{MF_S(v)/M_S}$$

[4]We used the Japanese morphological analyzer MeCab. (http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html)

T:	Chateau	Talbot	is	a	famous	winery	in	France	.
A:	O	O	O	O	O	O	O	S-PA	O
G:	B-PR	E-PR	O	O	O	O	O	S-PA	O

Figure 4: Example of a sentence with missing annotations. The row **T** shows the tokens of the sentence, the row **A** shows automatic annotations, and the row **G** shows golden annotations. The 'PR' and 'PA' are abbreviations of *Producer* and *Production Area*, respectively. Label 'O' means that a token is not annotated with any label.

［NUMBER］［ML (milliliter)］
［シャトー (Chateau)］［ANY］
［LOCATION］［ANY］［LOCATION］［市 (city)］

Figure 5: Examples of morpheme patterns induced from the KB for the wine category. The token [ANY] matches a sequence consisting of 1 - 3 arbitrary tokens. Tokens [LOCATION] and [NUMBER] are matched tokens whose part-of-speech is location and number, respectively.

where $MF_D(v)$ and $MF_S(v)$ are the MFs of value v in the product descriptions and structured data, respectively, and N_M is the number of merchants offering products in the category, and M_S is the number of merchants providing structured data in the category. The scoring function is designed so that values with a low MF for structured data and a high MF for item descriptions obtain high scores. We regard attribute values with scores greater than 30 as incorrect, and remove sentences that include annotation based on incorrect values from corpora. The threshold value was decided empirically.

4.2.3 Missing Annotation Filtering

Because of the small coverage of the KB, sentences with missing annotations are contained in the corpus. For example, although the tokens *Chateau* and *Talbot* in Figure 4 should be annotated as B-PRODUCER and E-PRODUCER[5] respectively, they are not annotated. These missing annotations result in reduced performance (especially *recall*) of models based on the corpus since they are considered to be *other* examples when training the models. One of the possible way to reduce the influence of the missing annotations is to discard sentences with missing annotations. Thus, we removed such sentences from the corpus.

To detect missing annotations, we generate morpheme patterns from values in the KB. Examples of the patterns for the wine category are given in Figure 5. First, attribute values are tokenized using a morphological analyzer, and then the PrefixSpan algorithm (Pei et al., 2001) is executed on the tokenized result to induce morpheme patterns[6]. We employed patterns that do not start and end with the [ANY] token, and which match attribute values in the KB and the total number of merchants corresponding to the matched values is greater than one.

[5]The beginning and the end of a value for the attribute called "Producer."

Table 2: Features for training extraction models.

Basic features (BFs)	
Feature	**Description**
Token	Surface form of the token.
Base	Base form of the token.
PoS	Part-of-speech tag of the token.
Char. types	Types of characters contained in the token. We defined *Hiragana*, *Katakana*, *Kanji*, *Alphabet*, *Number*, and *Other*.
Prefix	Double character prefix of the token.
Suffix	Double character suffix of the token.
Context features	
Feature	**Description**
Context	BFs of ± 3 tokens surrounding the token.

4.3 Extraction Model Training

Models for attribute value extraction are trained using the corpus. We employed Conditional Random Fields (CRFs), and used CRFsuite[7] with default parameter settings as the implementation thereof. The features we used are listed in Table 2.

We built a single model for each category. In other words, we did not build a separate model for each attribute of each category.

5 Experiments

This section reports the experimental results. We carried out the experiments on eight categories, and evaluated them using the dataset discussed in Section 5.1. An evaluation was conducted for each component, that is, evaluation of the KB (Section 5.2), evaluation of the automatically annotated corpora (Section 5.3), and evaluation of the extraction models (Section 5.4).

5.1 Construction of Evaluation Dataset

We created an evaluation dataset comprising 1,776 product pages gathered from the eight categories in Table 3. In constructing the dataset, for each category, we first listed top 300 merchants according to the number of products offered by the mer-

[6]We used PrefixSpan-rel as the implementation of PrefixSpan.(http://prefixspan-rel.sourceforge.jp/)

[7]http://www.chokkan.org/software/crfsuite/

Table 3: Categories and their selected attributes. The symbol $^{\#}$ denotes incorrect attributes. The symbol * is attached to the attributes that are not aggregated into their synonyms. For example, "country of origin" for wine is marked because it is not aggregated with "production area".

Category	Attributes (Each attribute is represented by one of its synonyms.)
Wine	容量 (volume), 品種 (grape variety), タイプ (type), 産地 (production area), アルコール度数 (alcohol), 原産国 (country of origin)*, 生産者 (producer), 原材料 (material)
T-shirt(men)	サイズ (size), 素材 (material), 色 (color), 着丈 (length)*, 身幅 (body width)*, M(M size)$^{\#}$, 肩幅 (shoulder width)*, L(L size)$^{\#}$
Printer ink	容量 (volume), サイズ (size), カラー (color), 重量 (weight), 色 (color)*, 適応機種 (compatible model), 材質 (material), 製造国 (production area)
Shampoo	容量 (volume), メーカー (manufacturer), 製造国 (production area), 成分 (constituent), 商品名 (product name), 区分 (category), サイズ (size), 重量 (weight)
Golf ball	コア (core), サイズ (size), カバー (cover), 材質 (material), 重さ (weight), 原産国 (country of origin), ディンプル形状 (shape of dimple), 色 (color)
Video game	サイズ (size), 重さ (weight), 材質 (material), 付属品 (accessory), 製造国 (production country), 色 (color), 対応機種 (compatible model), ケーブル長 (length of cable)
Car spotlight	サイズ (size), 色温度 (color temperature), 色 (color), 材質 (material), 重量 (weight), 適合車種 (compatible model), 製造国 (production country), 品番 (part number)
Cat food	内容量 (volume), 原産国 (production country), 粗繊維 (crude fiber), 粗脂肪 (crude fat), 粗灰分 (crude ash), 水分 (wet), 粗タンパク質 (crude protein), 重量 (weight)*

Table 4: Statistics of the corpora. $p^{\#}$, $s^{\#}$, and $v^{\#}$ denote the number of annotated pages, the number of annotated sentences, and the number of values, respectively.

Category	Test data (manually annotated)			Training data (automatically annotated)		
	$p^{\#}$	$s^{\#}$	$v^{\#}$	$p^{\#}$	$s^{\#}$	$v^{\#}$
Wine	282	1,863	3,040	25,358	28,952	48,645
T-shirt(men)	259	2,580	5,534	14,978	18,018	41,954
Printer ink	273	1,230	4,029	8,473	13,562	42,969
Shampoo	233	1,518	4,352	18,669	30,263	53,294
Golf ball	160	555	719	1,114	2,109	2,760
Video game	212	807	1,088	19,292	29,356	35,230
Car spotlight	271	1,401	2,282	8,124	12,910	18,937
Cat food	86	276	452	4,915	7,375	8,843
Total	1,776	10,230	19,496	100,923	142,545	252,632

chant in the category. Then, we randomly picked one from the product pages of each merchant. We extracted titles and sentences from the pages based on HTML tags. These texts were passed to the annotation process.

In selecting the attributes to be used for annotation, we selected the top eight attributes in each category according to the MFs of the attributes. Then, we manually discarded incorrect attributes and aggregated synonymous attributes that were not automatically discovered. These attributes are marked up with the symbols $^{\#}$ and * in Table 3, and are not considered in the evaluation in Sections 5.3 and 5.4.

An annotator was asked to annotate expressions in the text data, which could be considered as values of the selected attributes. The annotator was also asked to discard pages that offered multiple products and miscategorized products.

In this way, the evaluation dataset was constructed by one annotator. After the construction, we checked the accuracy of the annotation. We picked up 400 annotated pages, and then checked them by another annotator whether annotations in the pages were correct. The agreement of annotations between the two annotators was about 88.4%. Statistics of the dataset are given in the *Test data* column in Table 4.

5.2 Evaluation of Knowledge Base

5.2.1 Evaluation of Extracted Attributes

We checked whether attributes extracted using our method could be regarded as *correct*. We asked two subjects to judge 411 expressions, all extracted attributes for the eight categories. The ratio of attributes that were judged as *correct* by both annotators was 0.776. The kappa statistics between the annotators was 0.581. This value is defined as *moderate agreement* in (Landis and Koch, 1977). Majority of the attributes judged as *incorrect* were extracted from complex tables and tables on miscategorized pages.

5.2.2 Evaluation of Attribute Synonyms

Next, we assessed the performance of our synonym discovery. We asked the subjects to aggregate synonymous attributes in KBs, and then computed *purity* and *inverse purity* scores (Artiles et al., 2007) using the data. We discarded attributes judged as incorrect when computing these scores. We computed macro-averaged scores for each subject, and then averaged them.

As a result, the averaged purity and inverse purity were 0.920 and 0.813, respectively. The purity score is close to perfect, which means that the merged expressions are mostly regarded as synonyms. On the other hand, the score for in-

Table 5: Accuracy of KBs. # shows the total number of KB entries with each MF.

MF of pairs	Wine		Shampoo	
	#	Acc. [%]	#	Acc. [%]
≥ 1 (All)	3,940	75.3 (301/400)	6,798	67.5 (270/400)
≥ 2	751	97.2 (69/71)	2,307	90.8 (118/130)
≥ 3	384	97.1 (33/34)	1,543	97.3 (73/75)
≥ 5	215	95.5 (21/22)	931	98.1 (52/53)

Table 6: Accuracy of our annotation method.

Annotation method	Prec. (%)	Recall (%)	F_1 score
(1) Naive annotation	47.46	**45.48**	**46.19**
(2) (1) + incorrect	51.39	39.14	43.00
(3) (2) + missing	**57.14**	29.29	37.21

verse purity is less than 0.82. Improvement of the methodology in terms of coverage is left for future work.

5.2.3 Evaluation of Attribute Values

We evaluated the quality of the KBs for the wine and shampoo categories only, because the evaluation for all categories requires enormous human effort. We randomly selected 400 values of attributes listed in Table 3, and then asked the subjects to judge whether the values could be regarded as *correct* for the attributes. To judge the pair <*attr.*, *value*> in the KB for category C, we automatically generated the sentence: "*value* is an expression that represents (*attr.* or S^1_{attr} or ... or S^n_{attr}) of C." Here, S^n_{attr} denotes the nth synonym of *attr*. The subjects judged a pair to be *correct* if the sentence generated from the pair was naturally acceptable. For example, the pair <*variety*, *onion*> in the KB for the wine category is deemed incorrect because the sentence "*onion* is an expression that represents (*variety* or *grape variety* or *usage variety*) of *wine*." is not acceptable.

The evaluation results are given in Table 5. The kappa statistics between the annotators were 0.632 for the wine category, and 0.678 for the shampoo category, respectively. These values indicate *good agreement*. We regarded a pair as correct if the pair was judged as correct by both annotators. We can see that the accuracy of each category is promising. In particular, the accuracy of pairs with MF greater than one is 90% or more. This means that merchant frequency plays a crucial role in constructing KBs from structured data that are embedded in product pages by different merchants.

5.3 Evaluation of the Annotated Corpora

We also checked the effectiveness of the proposed annotation method by annotating the same product pages as those in the evaluation dataset, and then checking overlaps between the manual and automatic annotations. The experimental results are given in Table 6. We judged an extracted value to be *correct* if the value exactly matched the manually annotated one. The results are given as micro-averaged precision, recall, and F_1 scores for each attribute in each category. We can see that the proposed filtering methods improve the precision (annotation quality) at the expense of recall.

KB match: Naive KB matching for corpora (same as (3) in Table 6).

Model w/o filters: Training models based on corpora naively annotated using KBs. That is, filters for incorrect and missing annotations are not applied.

Model with incorrect annotation filter (Model with incorrect only): Training models based on corpora where only the filter for incorrect annotations is applied.

Model with missing annotation filter (Model with miss only): Training models based on corpora where only the filter for missing annotations is applied.

Figure 6: Alternative methods.

Table 7: Micro-averaged precision, recall and F score of the models for proposed method and alternatives.

Method	Prec.(%)	Recall (%)	F_1 score
Supervised Model	88.28	58.15	68.66
KB match	57.14	29.29	37.21
Model w/o filters	52.60	54.49	53.14
Model with incorrect only	**60.46**	54.23	56.84
Model with miss only	50.47	**59.71**	54.43
Model for proposed method	57.05	59.66	**58.15**

5.4 Evaluation of Extraction Model

We compared the performance of the extraction models trained for each category with the alternative methods shown in Figure 6. We naively matched entries in the KB for unlabeled product pages, and then randomly picked 100,000 unique sentences from the annotated pages. We refer to the picked sentences as the *Raw Corpus* (RC). Then, we ran the filters and training process on the RC since we were limited by the RAM required by CRFsuite. Some statistics of the corpora after applying the filters are shown in the column *Training data* in Table 4.

The evaluation results are shown in Table 7. **Model w/o filters** outperformed **KB match** by as much as 15.9 points in F_1 score. These improvements are caused by improving the recall of the method. This shows that contexts surrounding a value and patterns of tokens in a value are successfully captured. **Model with incorrect only** also achieved higher performance than **Model w/o**

Table 8: Types of errors.

Type		# err.
Automatic annotation	Context dependent role	15
	Part of a compound word	12
	Polysemous word	9
Incorrect KB entry		23
Over generation by	learned patterns	15
Extraction from	unrelated regions	12
Others		14

filters. Especially, the precision of the extraction models is improved by 7.9%. This means that the incorrect annotation filter successfully removed annotation based on incorrect KB entries from the annotated corpora. In addition, **Model with miss only** achieved higher performance than **Model w/o filters**. In particular, the recall of the method improved by 5.2%. This shows that the missing annotation filter effectively works for precisely training extraction models. As a result, the precision and recall of the proposed method are enhanced by employing both filters simultaneously, and the method achieved 58.15 points in F_1 scores.

By comparison, the performance of the extraction models based on manually annotated corpora is shown in Table 7. The supervised method was evaluated with 10-fold cross validation. From the table, we can see that the recall of our method outperforms that of the supervised method while the precision and F_1 score of our method are lower those of the supervised method.

6 Discussion

For the wine and shampoo categories, we randomly picked up 50 attribute values that were judged as incorrect. Then we classified them according to their error types. The classification result is shown in Table 8. Ratios of errors based on automatic annotation, and incorrect KB entities, over generation by learned patterns, and extraction from unrelated region with products are 36%, 23%, 15% and 12%, respectively.

The errors stemming from automatic annotation can be classified into three sub-types. Errors that require understanding of the context when annotating attribute values are the most common sub-type. For example, in the wine domain, the attribute-value pair <Production area, Bordeaux> was extracted from the following sentence:

- 土壌が ボルドー$_{<Productionarea>}$ のポムロールと非常に似ている。
  (Soil is very similar with ones in Pomerol region of Bordeaux$_{<Productionarea>}$.)

Although the extracted pair can be regarded as a correct KB entity for the wine categories, it is not *production areas* of wine in the above sentence. This type of error can be reduced if we can successfully leverage the above sentence as a negative example in the model training step. To generate such negative examples is future work.

The second type of errors left for future work occurs in annotation of compound words. For example, automatically annotated corpora for the shampoo category has the following sentence:

- **ヒアルロン酸**$_{<Constituent>}$ **以上**の保水力がある。
  (It has a higher water-holding ability **than hyaluronan**$_{<Constituent>}$ has.)

This sort of annotation errors may decrease if we omit the annotation of parts of compound words.

The third type of errors is annotation based on polysemous words. For instance, although <Alcohol, 10%> for the wine category is a correct KB entry, the word "10%" is used for describing various types of ratios. The following sentence is one of the examples where the word 10% is used with a meaning other than alcohol content in the wine domain:

- 輸出は全体の 10 %$_{<Alcohol>}$ 程度。
  (The amount of exports is approximately 10 %$_{<Alcohol>}$ of the total.)

A wrong extraction model for the alcohol attribute is trained through the above sentence. Disambiguation of attribute values is required in the annotation step in order to train precise models. On the task of extracting person names, a method for disambiguation of names is proposed by Bollegala et al. (2012). To employ similar disambiguation methodology is one of our plans for future work.

7 Conclusion

We proposed a purely unsupervised methodology for extracting attributes and their values from e-commerce product pages. We showed that the performance of our method attained an average F score of approximately 58.2 points using manually annotated corpora.

We believe the most important task for future work is to improve annotation quality. Disambiguation of attribute values and construction of wide coverage KBs are crucial to boost the quality. Another important future task concerns synonymy. We only tackled attribute synonymy. Discovery of attribute value synonyms is also an important future direction.

References

Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2007. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 64–69.

Lidong Bing, Tak-Lam Wong, and Wai Lam. 2012. Unsupervised extraction of popular product attributes from web sites. In *Proceedings of the Eighth Asia Information Retrieval Societies Conference*, pages 437–446.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2012. Automatic annotation of ambiguous personal names on the web. *Computational Intelligence*, 28(3):398–425.

Nilesh N. Dalvi, Philip Bohannon, and Fei Sha. 2009. Robust web extraction: an approach based on a probabilistic tree-edit model. In *Proceedings of the 2009 ACM International Conference on Management of Data*, pages 335–348.

Remi Ferrez, Clement Groc, and Javier Couto. 2013. Mining product features from the web: A self-supervised approach. In *Web Information Systems and Technologies*, volume 140 of *Lecture Notes in Business Information Processing*, pages 296–311.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48.

Pankaj Gulhane, Rajeev Rastogi, Srinivasan H. Sengamedu, and Ashwin Tengli. 2010. Exploiting content redundancy for web information extraction. In *Proceedings of the 19th International World Wide Web Conference*, pages 1105–1106.

Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 407–415.

J. Richard Landis and Gary. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Karin Mauge, Khash Rohanimanesh, and Jean-David Ruvini. 2012. Structuring e-commerce inventory. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 805–814.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the Fourth International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining*, pages 577–583.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th IEEE International Conference of Data Engineering*, pages 215–224.

Katharina Probst, Rayid Ghani, Marko Krema, Andrew Fano, and Yan Liu. 2007. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of the 20th International Joint Conference in Artificial Intelligence*, pages 2838–2843.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 104–111.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A decision tree method for finding and classifying names in japanese texts. In *In Proceedings of the Sixth Workshop on Very Large Corpora*.

Keiji Shinzato and Kentaro Torisawa. 2004. Acquiring hyponymy relations from web documents. In *Proceedings of Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting*, pages 73–80.

Kentaro Torisawa. 2001. An unsupervised method for canonicalization of japanese postpositions. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, pages 211–218.

Tak-Lam Wong, Wai Lam, and Tik-Shun Wong. 2008. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the 31st ACM SIGIR Conference*, pages 35–42.

Naoki Yoshinaga and Kentaro Torisawa. 2006. Finding specification pages according to attributes. In *Proceedings of the 15th International World Wide Web Conference*, pages 1021–1022.