Structuring E-Commerce Inventory

Karin Mauge
eBay Research Labs
2145 Hamilton Avenue
San Jose, CA 95125
kmauge@ebay.com

Khash Rohanimanesh
eBay Research Labs
2145 Hamilton Avenue
San Jose, CA 95125
krohanimanesh@ebay.com

Jean-David Ruvini
eBay Research Labs
2145 Hamilton Avenue
San Jose, CA 95125
jruvini@ebay.com

Abstract

Large e-commerce enterprises feature millions of items entered daily by a large variety of sellers. While some sellers provide rich, structured descriptions of their items, a vast majority of them provide unstructured natural language descriptions. In the paper we present a 2 steps method for structuring items into descriptive properties. The first step consists in unsupervised property discovery and extraction. The second step involves supervised property synonym discovery using a maximum entropy based clustering algorithm. We evaluate our method on a year worth of e-commerce data and show that it achieves excellent precision with good recall.

1 Introduction

Online commerce has gained a lot of popularity over the past decade. Large on-line C2C marketplaces like eBay and Amazon, feature a very large and long-tail inventory with millions of *items* (product offers) entered into the marketplace every day by a large variety of sellers. While some sellers (generally large professional ones) provide rich, structured description of their products (using schemas or via a global trade item number), the vast majority only provide unstructured natural language descriptions.

To manage items effectively and provide the best user experience, it is critical for these marketplaces to structure their inventory into descriptive name-value pairs (called properties) and ensure that items of the same kind (digital cameras for instance) are described using a unique set of property names (brand, model, zoom, resolution, etc.) and values. For example, this is important for measuring item similarity and complementarity in merchandising, providing faceted navigation and various business intelligence applications. Note that structuring items does not necessarily mean identifying products as not all e-commerce inventory is manufactured (animals for examples).

Structuring inventory in the e-commerce domain raises several challenges. First, one needs to identify and extract the names and the values used by individual sellers from unstructured textual descriptions. Second, different sellers may describe the same product in very different ways, using different terminologies. For example, Figure 1 shows 3 item descriptions of hard drives from 3 different sellers. The left description mentions ”rotational speed” in a specification table while the other two descriptions use the synonym ”spindle speed” in a bulleted list (top right) or natural language specifications (bottom right). This requires discovering semantically equivalent property names and values across inventories from multiple sellers. Third, the scale at which on-line marketplaces operate makes impractical to solve any of these problems manually. For instance, eBay reported 99 million active users in 2011, many of whom are sellers, which may translate into thousands or even millions of synonyms to discover accross more than 20,000 categories ranging from consumer electronics to collectible and art.

This paper describes a two step process for structuring items in the e-commerce domain. The first step consists in an unsupervised property extraction technique which allows discovering name-value

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 805–814,
Jeju, Republic of Korea, 8-14 July 2012.

pairs from unstructured item descriptions. The second step consists in identifying semantically equivalent property names amongst these extracted properties. This is accomplished using supervised maximum entropy based clustering. Note that, although value synonym discovery is an equally important task for structuring items, this is still an area of ongoing research and is not addressed in this paper.

The remainder of this paper is structured as follows. We first review related work. We then describe the two steps of our approach: 1) unsupervised property discovery and extraction and 2) property name synonym discovery. Finally, we present experimental results on real large-scale e-commerce data.

2 Related Work

This section reviews related work for the two components of our method, namely unsupervised property extraction and supervised property name synonym discovery.

2.1 Unsupervised Property Extraction

A lot of progress has been accomplished in the area of property discovery from product reviews since the pioneering work by (Hu and Liu, 2004). Most of this work is based on the observation, later formalized as *double propagation* by (Qiu et al., 2009), that in reviews, opinion words are usually associated with product properties in some ways, and thus product properties can be identified from opinion words and opinion words from properties alternately and iteratively. While (Hu and Liu, 2004) initially used association mining techniques; (Liu et al., 2005) used Part-Of-Speech and supervised rule mining to generate language patterns and identify product properties; (Popescu and Etzioni, 2005) used point wise mutual information between candidate properties and meronymy discriminators; (Zhuang et al., 2006; Qiu et al., 2009) improved on previous work by using dependency parsing; (Kobayashi et al., 2007) mined property-opinion patterns using statistical and contextual cues; (Wang and Wang, 2008) leveraged property-opinion mutual information and linguistic rules to identify infrequent properties; and (Zhang et al., 2010) proposed a ranking scheme to improve double propagation precision. In this paper, we are focusing on extracting properties from product descriptions which do not contain opinion words.

In a sense, item properties can be viewed as slots of product templates and our work bears similarities with template induction methods. (Chambers and Jurafsky, 2011) proposed a method for inferring event templates based on word clustering according to their proximity in the corpus and syntactic function clustering. Unfortunately, this technique cannot be applied to our problem due to the lack of discourse redundancy within item descriptions.

(Putthividhya and Hu, 2011) and (Sachan et al., 2011) also addressed the problem of structuring items in the e-commerce domain. However, these works assume that property names are known in advance and focus on discovering values for these properties from very short product titles.

Although we are primarily concerned with unsupervised property discovery, it is worth mentioning (Peng and McCallum, 2004) and (Ghani et al., 2006) who approached the problem using supervised machine learning techniques and require labeled data.

2.2 Property Name Synonym Discovery

Our work is related to the synonym discovery research which aims at identifying groups of words that are semantically identical based on some defined similarity metric. The body of work on this problem can be divided into two major approaches (Agirre et al., 2009): methods that are based on the available knowledge resources (e.g., WordNet, or available taxonomies) (Yang and Powers, 2005; Alvarez and Lim, 2007; Hughes and Ramage,), and methods that use contextual/property distribution around the words (Pereira et al., 1993; Chen et al., 2006; Sahami and Heilman, 2006; Pantel et al., 2009). (Zhai et al., 2010) propose a constrained semi-supervised learning method using a naive Bayes formulation of EM seeded by a small set of labeled data and a set of soft constraints based on the prior knowledge of the problem. There has been also some recent work on applying topic modeling (e.g., LDA) for solving this problem (Guo et al., 2009).

Our work is also related to the existing research on schema matching problem where the objective is to identify objects that are semantically related cross schemas. There has been an extensive study on the

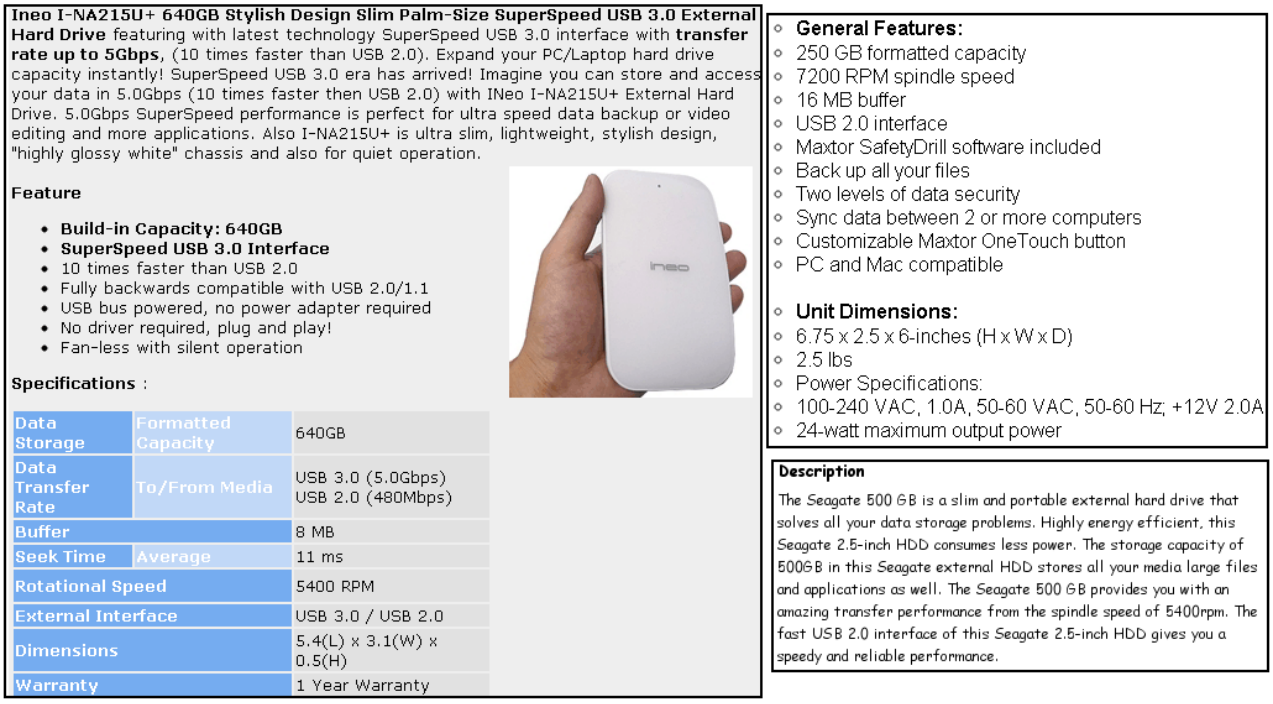

Figure 1: Three examples of item descriptions containing a specification table (left image), a bulleted list (top right) and natural language specifications (bottom right).

problem of schema matching (for a comprehensive survey see (Rahm and Bernstein, 2001; Bellahsene et al., 2011; Bernstein et al., 2011)). In general the work can be classified into rule-based and learning-based approaches. Rule-based systems (Castano and de Antonellis, 1999; Milo and Zohar, 1998; L. Palopol and Ursino, 1998) often utilize only the schema information (e.g., elements, domain types of schema elements, and schema structure) to define a similarity metric for performing matching among the schema elements in a hard coded fashion. In contrast learning based approaches learn a similarity metric based on both the schema information and the data. Earlier learning based systems (Li and Clifton, 2000; Perkowitz and Etzioni, 1995; Clifton et al., 1997) often rely on one type of learning (e.g., schema meta-data, statistics of the data content, properties of the objects shared between the schemas, etc). These systems do not exploit the complete textual information in the data content therefore have limited applicability. Most recent systems attempt to incorporate the textual contents of the data sources into the system. Doan et al. (2001) introduce *LSD* which is a semi-automatic machine learning based matching framework that trains a set of base learners using a set of user provided semantic mappings over a small data sources. Each base learner exploits a different type of information, e.g. source schema information and information in the data source. Given a new data source, the base learners are used to discover semantic mappings and their prediction is combined using a meta-learner. Similar to LSD, GLUE (Doan et al., 2003) also uses a set of base learners combined into a meta-learner for solving the matching problem between two ontologies. Our work is mostly related to (Wick et al., 2008) where they propose a general framework for performing jointly schema matching, co-reference and canonicalization using a supervised machine learning approach. In this approach the matching problem is treated as a clustering problem in the schema attribute space, where a cluster captures a matched set of attributes. A conditional random field (CRF) (Lafferty et al., 2001) is trained using user provided mappings between example schemas, or ontologies. CRF bene-

fits from first order logic features that capture both schema/ontology information as well as textual features in the related data sources.

3 Unsupervised Property Extraction

The first step of our solution to structuring e-commerce inventory aims at discovering and extracting relevant properties from items.

Our method is unsupervised and requires no prior knowledge of relevant properties or any domain knowledge as it operates the exact same way for all items and categories. It maintains a set of previously discovered properties called *known properties* with *popularity* information. The popularity of a given property name N (resp. value V) is defined as the number of sellers who are using N (resp. V). A seller is said to use a name or a value if we are able to extract the property name or value from at least one of its item descriptions. The method is incremental in that it starts with an empty set of known properties, mines individual items independently and incrementally builds and updates the set of known properties.

The key intuition is that the abundance of data in e-commerce allows simple and scalable heuristic to perform very well. For property extraction this translates into the following observation: although we may need complex natural language processing for extracting properties from each and every item, simple patterns can extract most of the relevant properties from a subset of the items due to redundancy between sellers. In other words, popular properties are used by many sellers and some of them write their descriptions in a manner that makes these properties easy to extract. For example one pattern that some sellers use to describe product properties often starts by a property name followed by a colon and then the property value (we refer to this pattern as the *colon pattern*). Using this pattern we can mine colon separated short strings like "size : 20 inches" or "color : light blue" which enables us to discover most relevant property names. However, such a pattern extracts properties from a fraction of the inventory only and does not suffice. We are using 4 patterns which are formally defined in Table 1.

All patterns run on the entire item description. Pattern 1 skips the html markers and scripts and applies only to the content sentences. It ignores any candidate property which name is longer than 30 characters and values longer than 80 characters. These length thresholds may be domain dependent. They have been chosen empirically. Pattern 2, 3 and 4 search for known property names. Pattern 2 extracts the closest value to the right of the name. It allows the name and the value to be separated by special characters or some html markups (like "<TR>", "<TD>", etc.). It captures a wide range of name value pair occurrences including rows of specification tables.

Syntactic cleaning and validation is performed on all the extracted properties. Cleaning consists mainly in removing bullets from the beginning of names and punctuation at the end of names and values. Validation rejects properties which names are pure numbers, properties that contain some special characters and names which are less than 3 characters long. All discovered properties are added to the set of known properties and their popularity counts are updated.

Note that for efficiency reasons, Part-Of-Speech (POS) tagging is performed only on sentences containing the *anchor* of a pattern. The anchor of pattern 1 is the colon sign while the anchor of the other patterns is the known property name KN. We use (Toutanova et al., 2003) for POS tagging.

4 Property Synonym Discovery

In this section we briefly overview a probabilistic pairwise property synonym model inspired by (Culotta et al., 2007).

4.1 Probabilistic Model

Given a category $\mathcal{C}$, let $\mathcal{X}_\mathcal{C} = \{x_1, x_2, \ldots, x_n\}$ be the raw set of n property names (prior to synonym discovery) extracted from a corpus of data associated with that category. Every property name is associated with pairs of values and popularity count (as defined in Section 3) $\mathcal{V}_{x_i} = \{\langle v_j^i, c^i(v_j^i)\rangle\}_{j=1}^m$, where v_j^i is the j^{th} value associated for the property name x_i and $c^i(v_j^i)$ is the popularity of value v_j^i. Given a pair of property names $x_{ij} = \{x_i, x_j\}$, let the binary random variable y_{ij} be 1 if x_i and x_j are synonyms. Let $\mathcal{F} = \{f_k(x_{ij}, y)\}$ be a set of features over x_{ij}. For example, $f_k(x_{ij}, y)$ may indicate

#	Pattern	Example
1	`[NP][:][optional DT][NP]`	”color : light blue”
2	`[KN][optional html][NP]`	”size</TD><TD><FONT COLOR="red">20 inches”
3	`[!IN][KN]["is" or "are"][NP]`	”color is red”
4	`[NP][KN]`	”red color”

Table 1: Patterns used to extract properties from item description. The macro tag NP denotes any of the tags NN, NNP, NNS, NNPS, JJ, JJS or CD. The KN tag is defined as a NP tag over a known property name. Pattern 1 only can discover new names; patterns 2 to 4 aim at capturing values for known property names.

whether x_i and x_j have both numerical values. Each feature f_k has an associated real-valued parameter λ_k. The *pairwise* model is given by:

$$\mathcal{P}(y_{ij}|x_{ij}) = \frac{1}{Z_{x_{ij}}} \exp \sum_k \lambda_k f_k(x_{ij}, y_{ij}) \qquad (1)$$

where $Z_{x_{ij}}$ is a normalizer that sums over the two settings of y_{ij}. This is a maximum-entropy classifier (i.e. logistic regression) in which $\mathcal{P}(y_{ij}|x_{ij})$ is the probability that x_i and x_j are synonyms. To estimate $\Lambda = \{\lambda_k\}$ from labeled training data, we perform gradient ascent to maximize the log-likelihood of the labeled data.

Given a data set in which property names are manually clustered, the training data can be created by simply enumerating over each pair of synonym property names x_{ij}, where y_{ij} is true if x_i and x_j are in the same cluster. More practically, given the raw set of extracted properties, first we manually cluster them. Positive examples are then pairs of property names from the same cluster. Negative examples are pairs of names cross two different clusters randomly selected. For example, let assume that the following four property name clusters have been constructed: $\{color, shade\}$, $\{size, dimension\}$, $\{weight\}$, $\{features\}$. These clusters implies that ”color” and ”shade” are synonym; that ”size” and ”dimension” are synonym and that ”weight” and ”features” don't have any synonym. The pair $(color, shade)$ is a positive examples, while $(size, shade)$ and $(weight, features)$ are negative examples.

Now, given an unseen category $\mathcal{C}'$ and the set of raw properties (property names and values) mined from that category, we can construct an undirected-weighted graph in which vertices correspond to the property names $\mathcal{N}_{\mathcal{C}'}$ and edge weights are proportional to $\mathcal{P}(y_{ij}|x_{ij})$. The problem is now reduced to finding the maximum a posteriori (MAP) setting of $y_{ij}s$ in the new graph. The inference in such models is generally intractable, therefore we apply approximate graph partitioning methods where we partition the graph into clusters with high intra-cluster edge weights and low inter-cluster edge weights. In this work we employ the standard *greedy agglomerative clustering*, in which each noun phrase would be assigned to the most probable cluster according to $\mathcal{P}(y_{ij}|x_{ij})$.

4.2 Features

Given a pair of property names $x_{ij} = \{x_i, x_j\}$ we have designed a set of features as follows:

Property name string similarity/distance: This measures string similarity between two names. We have included various string edit distances such as *Jaccard* distance over *n-grams* extracted from the property names, and also *Levenstein* distance. We have also included a feature that compares the two property names after their commoner morphological and inflectional endings have been removed using the *Porter Stemmer* algorithm.

Property value set coverage: We compute a weighted *Jaccard* measure given the values and the value frequencies associated with a property name.

$$\mathcal{J}(\mathcal{V}_{x_i}, \mathcal{V}_{x_j}) = \frac{\sum_{v \in (\mathcal{V}_{x_i} \cap \mathcal{V}_{x_j})} \min(c^i(v), c^j(v))}{\sum_{v \in (\mathcal{V}_{x_i} \cap \mathcal{V}_{x_j})} \max(c^i(v), c^j(v))}$$

This feature essentially computes how many property values are common between the two property names, weighted by their popularity.

Property name co-occurrence: This is an interesting feature which is based on the observation that

two property names that are synonyms, rarely occur together within the same description. This is based on the assumption that sellers are consistent when using property names throughout a single description. For example when they are specifying the size of an item, they either use *size* or *dimensions* exclusively in a single description. However, it is more likely that two property names that are not synonyms appear together within a single description. To conform this assumption, we ran a separate experiment that measures the co-occurrence frequency of the property names in a single category. Table 2 shows a measurement of pairwise co-occurrence of a few example property names computed over the *Audio books* eBay category. Given a property name x let $\mathcal{I}(x)$ be the total number of descriptions that contain the name x. Now, given two property names x_i and x_j, we define a measure of co-occurrence of these names as:

$$\text{CO}(x_i, x_j) = \frac{\mathcal{I}(x_i) \cap \mathcal{I}(x_j)}{\mathcal{I}(x_i) \cup \mathcal{I}(x_j)}$$

In Table 2 it can be seen that synonym property names such as *"author"* and *"by"* have a zero co-occurrence measure, while semantically different property names such as *"format"* and *"read by"* have a non-zero co-occurrence measure.

5 Experimental results

This section presents experimental results on a real dataset. We first describe the dataset used for these experiments and then provide results for property extraction and property name synonym discovery.

5.1 Data set and methodology

All the results we are reporting in this paper were obtained from a dataset of several billion descriptions corresponding to a year worth of eBay item (no sampling was performed).

For listing an item on eBay, a seller must provide a short descriptive title (up to 80 characters) and can optionally provide a few descriptive name value pairs called item specifics, and a free-form html description. Contrary to item specifics, a vast majority of sellers provide a rich description containing very useful information about the property of their item. Figure 1 shows 3 examples of eBay descriptions.

eBay organizes items into a six-level category structure similar to a topic hierarchy comprising 20,000 leaf categories and covering most of the goods in the world. An item is typically listed in one category but some items may be suitable for and listed in two categories.

Although this dataset is not publicly available, very similar data can be obtained from the eBay web site and through eBay Developers API [1].

In the following, we report precision and recall results. Evaluation was performed by two annotators (non expert of the domain). For property extraction, they were asked to decide whether or not an extracted property is relevant for the corresponding items; for synonym discovery to decide whether or not sellers refer to the same semantic entity. Annotators were asked to reject the null hypothesis only beyond reasonable doubt and we found the annotator agreement to be extremely high.

5.2 Property Extraction Results

We have been running the property extraction method described in Section 3 on our entire dataset. The properties extracted have been aggregated at the leaf category level and ranked by popularity (as defined in Section 3). Because no gold standard data is available for this task, evaluation has to be performed manually. However, it is impractical to review results for 20,000 categories and we uniformly sampled 20 categories randomly.

Precision. Table 3 shows the weighted (by category size) average precision of the extracted property names up to rank 20. Precision at rank k for a given category is defined as the number of relevant properties in the top k properties of that categories, divided by k. Table 4 shows the top 15 properties extracted for five eBay categories.

Although we did not formally evaluate the precision of the discovered values, informal reviews have shown that this method extracts good quality values. Examples are "n/a", "well", "storage or well", "would be by well" and "by well" for the property name "Water" in the Land category; "metal", "plastic", "nylon", "acetate" and "durable o matter" for "Frame material" in Sunglasses; or "acrylic",

[1] See `https://www.x.com/developers/ebay/` for details.

	author	by	read by	format	narrated by
author		0	0.06	0.06	0.006
by	0		0.17	0.005	0.013
read by	0.06	0.17		0.035	0
format	0.06	0.005	0.035		0.006
narrated by	0.006	0.013	0	0.006	

Table 2: Co-occurrence measure computed over a subset of property names in the *Audio books* category. Some synonym property names such as *author* and *by* have zero co-occurrence frequency, while semantically different property names such as *format* and *read by* sometimes appear together in some of the item descriptions.

Rank	1	2	3	4	5	6	7	8	9	10
Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.992	0.992	0.986
Rank	11	12	13	14	15	16	17	18	19	20
Precision	0.986	0.997	0.986	1	0.998	1	1	0.959	0.722	0.747

Table 3: Weighted average precision of the top 20 extracted property names.

”oil”, ”acrylic on canvas” and ”oil on canvas” for ”Medium” in Paintings.

Sets of values tend to contain more synonyms than names. Also, we observed that some names exhibit polysemy issues in that their values clearly belong to several semantic clusters. An example of polysemy is the name ”Postmark” in the ”Postcards” categories which contains values like ”none, postally used, no, unused” and years (”1909, 1908, 1910...”). Cleaning and normalizing values is ongoing research effort.

Recall. Evaluating recall of our method requires comparing for each category, the number of relevant properties extracted to the number of relevant properties the descriptions in this category contain. It is dauntingly expensive. As a proxy for name recall, we examined 20 categories and found that our method discovered all the relevant popular property names.

It is quite remarkable that an unsupervised method like ours achieves results of that quality and is able to cover most of the good of the world with descriptive properties. To our knowledge, this has never been accomplished before in the e-commerce domain.

5.3 Synonym discovery results

To train our name synonym discovery algorithm, we manually clustered properties from 27 randomly selected categories as described in Section 4. This resulted in 178 clusters, 113 of them containing a single property (no synonym) and 65 containing 2 or more properties and capturing actual synonym information. Note that although estimating the co-occurrence table (see Table 2) can be computationally expensive, it is very manageable for such a small set of clusters. Scalability issues due to the large number of eBay categories (nearly 20,000) made impractical to use the solutions proposed in the past to solve that problem as baselines.

Results were produced by applying the trained model to the top 20 discovered properties for each and every eBay categories. The algorithm discovered 10672 synonyms spanning 2957 categories.

Precision. To measure the precision of our algorithm, we manually labeled 6618 synonyms as *correct* or *incorrect*. 6076 synonyms were found to be correct and 542 incorrect, a precision of 91.8%. Table 5 shows examples of synonyms and one of the categories where they have been discovered. Some of them are very category specific. For instance, while ”hp” means ”horsepower” for air compressors, it is an acronym of a well known brand in consumer electronics.

Recall. Evaluating recall is a more labor intensive task as it involves comparing, for each of the 2957 categories, the number of synonyms discovered to the number of synonyms the category con-

Land	Aquariums	iPod & MP3 Players	Acoustic Guitars	Postcards
State	Dimensions	Weight	Top	Condition
Zoning	Height	Width	Scale length	Publisher
County	Size	Depth	Neck	Size
Water	Width	Height	Bridge	Postmark
Location	Includes	Color	Finish	Postally used
Taxes	Weight	Battery type	Rosette	Type
Size	Depth	Dimensions	Binding	Age
Sewer	Capacity	Frequency response	Fingerboard	Stamp
Power	Color	Storage capacity	Tuning machines	Date
Roads	Power	Display	Case	Title
Lot size	LCD size	Capacity	Pickguard	Postmarked
Utilities	Length	Screen size	Tuners	Subject
Parcel number	Material	Battery	Nut width	Location
	Cable length	Length		Corners
	Condition	Thickness		Era

Table 4: Examples of discovered properties for 5 eBay categories.

Category	Synonyms
Rechargeable Batteries	{Battery type, Chemical composition}
Lodging	{Check-in, Check-in time}
Flower seeds	{Bloom time, Flowering season}
Doors & Door Hardware	{Colour,Color, Main color}
Gemstone	{Cut, Shape}
Air Compressors	{Hp, Horsepower}
Decorative Collectibles	{Item no, Item sku, Item number}
Router Memory	{Memory (ram), Memory size}
Equestrian Clothing	{Bust, Chest}
Traiding Cards	{Rarity, Availability}
Paper Calendar	{Time period, Calendars era}

Table 5: Examples of discovered property name synonyms.

tains. As a proxy we labeled 40 randomly selected categories. For these categories, we found the recall to be 51%. As explained in Section 4, the overlap of values between two names is an important feature for our algorithm. The fact that we are not cleaning and normalizing the values discovered by our property extraction algorithm clearly impacts recall. This is definitively an important direction for further improvements.

6 Conclusion

We presented a method for structuring e-commerce inventory into descriptive properties. This method is based on unsupervised property discovery and extraction from unstructured item descriptions, and on property name synonym discovery achieved using a supervised maximum entropy based clustering algorithm. Experiments on a large real e-commerce dataset showed that both techniques achieve very good results. However, we did not address the issue of property value cleaning and normalization. This is an important direction for future work.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marco A. Alvarez and SeungJin Lim. 2007. A graph modeling of semantic similarity between words. In *Proceedings of the International Conference on Semantic Computing*, pages 355–362, Washington, DC, USA. IEEE Computer Society.

Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors. 2011. *Schema Matching and Mapping*. Springer.

Philip A. Bernstein, Jayant Madhavan, and Erhard Rahm. 2011. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11):695–701.

Silvana Castano and Valeria de Antonellis. 1999. A schema analysis and reconciliation tool environment for heterogeneous databases. In *Proceedings of the 1999 International Symposium on Database Engineering & Applications*, IDEAS '99, pages 53–, Washington, DC, USA. IEEE Computer Society.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 976–986, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hsin-Hsi Chen, Ming-Shun Lin, and Yu-Chuan Wei. 2006. Novel association measures using web search with double checking. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 1009–1016, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Clifton, Ed Housman, and Arnon Rosenthal. 1997. Experience with a combined approach to attribute-matching across heterogeneous databases. In *In Proc. of the IFIP Working Conference on Data Semantics (DS-7*.

Aron Culotta, Michael Wick, Robert Hall, and Andrew Mccallum. 2007. First-order probabilistic models for coreference resolution. In *In Proceedings of HLT-NAACL 2007*.

AnHai Doan, Pedro Domingos, and Alon Y. Halevy. 2001. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, SIGMOD '01, pages 509–520, New York, NY, USA. ACM.

AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevy. 2003. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12:303–319, November.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8:41–48, June.

Honglei Guo, Huijia Zhu, Zhili Guo, XiaoXun Zhang, and Zhong Su. 2009. Product feature categorization with multilevel latent semantic association. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1087–1096, New York, NY, USA. ACM.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Thad Hughes and Daniel Ramage. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 581–589.

Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL.*

D. Saccà L. Palopol and D. Ursino. 1998. Semi-automatic, semantic discovery of properties from database schemes. In *Proceedings of the 1998 International Symposium on Database Engineering & Applications*, pages 244–, Washington, DC, USA. IEEE Computer Society.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wen-Syan Li and Chris Clifton. 2000. Semint: a tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data Knowl. Eng.*, 33:49–84, April.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions

on the web. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA. ACM.

Tova Milo and Sagit Zohar. 1998. Using schema matching to simplify heterogeneous data translation. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 122–133, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 938–947, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fuchun Peng and Andrew McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL04*, pages 329–336.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, pages 183–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mike Perkowitz and Oren Etzioni. 1995. Category translation: learning to understand information on the internet. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, pages 930–936, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA. Association for Computational Linguistics.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *EMNLP*, pages 1557–1567.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1199–1204, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350.

Mrinmaya Sachan, Tanveer Faruquie, L. V. Subramaniam, and Mukesh Mohania. 2011. Using text reviews for product entity completion. In *Poster at the 5th International Joint Conference on Natural Language Processing*, IJCNLP'11, pages 983–991.

Mehran Sahami and Timothy D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 377–386, New York, NY, USA. ACM.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Wang and Houfeng Wang. 2008. Bootstrapping both product features and opinion words from chinese customer reviews with cross-inducing. In *Proceedings of the Third International Joint Conference on Natural Language Processing*.

Michael L. Wick, Khashayar Rohanimanesh, Karl Schultz, and Andrew McCallum. 2008. A unified approach for schema matching, coreference and canonicalization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 722–730, New York, NY, USA. ACM.

Dongqiang Yang and David M. W. Powers. 2005. Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38*, ACSC '05, pages 315–322, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2010. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1272–1280, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1462–1470, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50, New York, NY, USA. ACM.