Extracting Facets from Lost Fine-Grained Categorizations in Dataspaces

Riccardo Porrini[1,2], Matteo Palmonari[1], and Carlo Batini[1]

[1] DISCo, University of Milano-Bicocca
Viale Sarca, 336/14, 20126, Milan, Italy
{matteo.palmonari,carlo.batini}@disco.unimib.it
[2] 7Pixel s.r.l.
Via Lanzoni 13, 27010, Giussago (PV), Italy
riccardo.porrini@trovaprezzi.it

Abstract. Categorization of instances in dataspaces is a difficult and time consuming task, usually performed by domain experts. In this paper we propose a semi-automatic approach to the extraction of facets for the fine-grained categorization of instances in dataspaces. We focus on the case where instances are categorized under heterogeneous taxonomies in several sources. Our approach leverages Taxonomy Layer Distance, a new metric based on structural analysis of source taxonomies, to support the identification of meaningful candidate facets. Once validated and refined by domain experts, the extracted facets provide a fine-grained classification of dataspace instances. We implemented and evaluated our approach in a real world dataspace in the eCommerce domain. Experimental results show that our approach is capable of extracting meaningful facets and that the new metric we propose for the structural analysis of source taxonomies outperforms other state-of-the-art metrics.

Keywords: dataspaces, web data integration, taxonomy integration, facet extraction.

1 Introduction

The *dataspace* abstraction describes data integration architectures that deal with large heterogeneous data, which are partially unstructured, possibly sparse and characterized by high dimensionality [6]. Differently from traditional data integration architectures, where data from local sources are consistently integrated in a global view after their schemas are aligned, pay-as-you-go data integration is needed in dataspaces: data are more and more integrated along time, as more effective data access features are required [11].

Data integration methodologies inspired by dataspace principles are widely adopted for industry-scale Web data integration because of the amount and heterogeneity of source instances to be integrated [5]. Several examples of Web data integration systems can be found in the eCommerce domain. *Price Comparison Engines* (PCEs) integrate a very large number of heterogeneous product *offers*

M. Jarke et al. (Eds.): CAiSE 2014, LNCS 8484, pp. 580–594, 2014.

(i.e., dataspace instances) from many different e-marketplaces providing search and browsing features over the integrated information. Through PCE front-ends, end-users compare different eMarketplaces product offerings in terms of price and/or product features. Many PCEs such as Google Shopping [1], PriceGrabber [2] and Amazon [3] have been developed, which differ in terms of coverage and effectiveness of search and browsing features.

Category-based and *facet-based* browsing are two examples of data access features that many PCEs aim to deliver to their users. These features require the creation and maintenance of categorizations respectively based on *taxonomies*, i.e., hierarchies of product categories such as "Mobile Phones" or "Wines", and *facets*, i.e., sets of mutually exclusive coordinate terms that belong to a same concept (e.g., "Grape: Barolo, ..., Cabernet", ..., "Type: Red Wine, ..., White Wine") [19,21]. A *global taxonomy* is used to annotate all the instances in the dataspace with a coarse-grained categorization, which helps end-users to rapidly recall the "family" of instances they are interested in. Facets can be used within a specific (global) taxonomy category, to annotate instances in the dataspace with a fine-grained categorization, which helps end-user to rapidly recall instances with specific characteristics (e.g., "Grape: Barolo", "Type: Red Wine"). Facet-based categorizations can be also useful in relation to Search Engine Optimization because facets indexed by search engines can bring more traffic to the PCEs' websites.

Unfortunately, facet creation and maintenance is an extremely time and effort consuming task in PCEs, which is left to manual work of domain experts. The definition of meaningful facets at large scale requires a deep understanding of the salient characteristics of dataspace instances (e.g., wines are characterized by grape, type, provenance, and so on) for a large number of diverse product categories. As a result, while a *global taxonomy* is used in most of PCEs[4], many PCEs provide only few generic facets (e.g., "Price Range" and "Merchant") and others provide a richer set of facets but only for a limited amount of popular product categories.

In this paper we propose an approach to automatic facet extraction in dataspaces, which is aimed to support domain experts in creating and maintaining significant facets associated with global categories. Our approach leverages the information already present in the dataspace, namely (i) taxonomies used to classify instances in the data sources and (ii) mappings established from source taxonomies to the global taxonomy of the dataspace, to suggest meaningful facets for a given global category. Unlike the global taxonomy, which has to cover instances from very diverse domains, source taxonomies are often specialized in certain domains (e.g., "Wines"). Domain experts map specific categories in source taxonomies (e.g., "Barolo") to generic categories in the global taxonomy (e.g., "Wines"). The idea behind our approach consists in reusing the

[1] `http://www.google.com/shopping`
[2] `www.pricegrabber.com/`
[3] `http://amazon.com/`
[4] See, e.g., `http://www.google.com/basepages/producttype/taxonomy.en-US.txt`

fine-grained categories that occur in several source taxonomies mapped to the global taxonomy (e.g., "Barolo","Cabernet"), to extract a set of relevant facets for a given global category (e.g., "Grape: Barolo, ..., Cabernet", ..., "Type: Red Wine, ..., White Wine").

Our approach incorporates an automatic facet extraction algorithm that consists of three steps: *extraction* of potential facet values (e.g., "Cabernet"); *clustering* of facet values into sets of mutually exclusive terms (e.g., "Bordeaux", "Cabernet", "Chianti"); *labeling* of clusters with meaningful labels (e.g., "Grape"). The algorithm is based on structural analysis of source taxonomies and on *Taxonomy Layer Distance*, a novel metric introduced to evaluate the distance between mutually exclusive categories in different taxonomies. Experiments conducted to evaluate the approach show that our algorithm is able to extract meaningful facets that can then be refined by domain experts. In addition, since our approach extracts facets from source categorizations, the annotation of the dataspace instances with the extracted facets is straightforward, supporting facet-based browsing.

The paper is organized as follows. The problem of facet extraction is defined and explained in Section 2. Our approach to facet extraction is described in Section 3 and the evaluation is discussed in Section 4. Related work is presented in Section 5. Section 6 draws conclusions and discusses future work.

2 Problem Definition and Domain Example

A *facet* can be defined as "a clearly defined, mutually exclusive, and collectively exhaustive aspect, property, or characteristic of a class or specific subject" [19]. As input to our problem, we assume that there exists a global taxonomy used in the dataspace and a set of mappings from leaf categories in source taxonomies to leaf global categories. We assume that the mappings have many-to-one cardinality, i.e., many categories in each source taxonomy are mapped one global category. Observe that mappings of this kind can be easily extracted in any dataspace where instances are categorized using one source taxonomy category and one global taxonomy category.

In the following paragraph we summarize the terminology used in the rest of the paper, along with a precise description of the problem of Facet Extraction.

Source Taxonomy: a source taxonomy consists of a partially ordered set S of source categories s.

Global Taxonomy: a global taxonomy consist of a partially ordered set G of global categories g.

Leaf-to-leaf Category Mapping: a leaf-to-leaf (leaf for brevity) category mapping $m : g \leftarrow s$ is a correspondence from a leaf category s of some source taxonomy S to a global leaf category g. The semantics of a leaf mapping from s to g is that instances that are classified under s at the source can be classified under g once they enter the dataspace.

Facet: a facet F^g for a global category g is a finite set of values $v_1, ..., v_n$ (e.g., {"Red Wine", "White Wine"}) associated with a label; conceptually, a facet

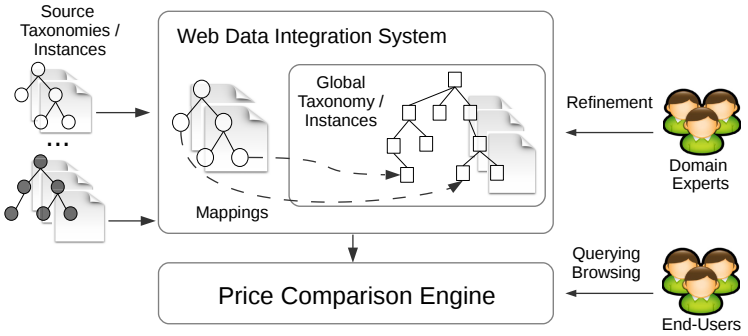

Fig. 1. Conceptual architecture of a Price Comparison Engine

label for a facet F^g briefly describes the concept of reality that facet values refer to (i.e., "Wine Type").

Facet Extraction Problem: given a global leaf category g, a set of mappings M from source categories $s_1, \ldots, s_n$ to g in the form $g \leftarrow s_1, \ldots, g \leftarrow s_n$, extract a set $\mathcal{F}^g$ of facets F^g, each one associated with a label.

As introduced in Section 1, the Facet Extraction problem is common in PCEs. The dataspace of a PCE consists of offers (i.e., the dataspace instances) coming from many eMarketplaces (i.e., the data sources). The conceptual architecture of a PCE is sketched in Figure 1. Each eMarketplace categorizes offers using a own source taxonomy. Source instances are integrated within the dataspace by specifying mappings from a large population of (often domain specific) source taxonomies to a global taxonomy. Mappings are defined and maintained by domain experts with the aid of (semi) automatic algorithms. To size the problem, we provide some figures about the dataspace of one of the most popular PCE on the Italian market. TrovaPrezzi[5] integrates many times per day 7 millions product offers from about 3900 eMarketplaces. Over more than 10 years of activity, more than 1 million of leaf mappings have been specified from source categories to more than 500 global categories.

3 Facet Extraction

The semi-automatic approach to facet extraction proposed in this paper is sketched in Figure 2 and is aimed to support domain experts who are in charge of maintaining classifications and mappings within a dataspace. Domain experts trigger the facet extraction process for a specified global category using a Web interface. An automatic facet extraction algorithm suggests a set of facets to domain experts, who inspect, validate and refine the result of the automatic extraction algorithm, deciding which facets will be part of the dataspace.

[5] `www.trovaprezzi.it`

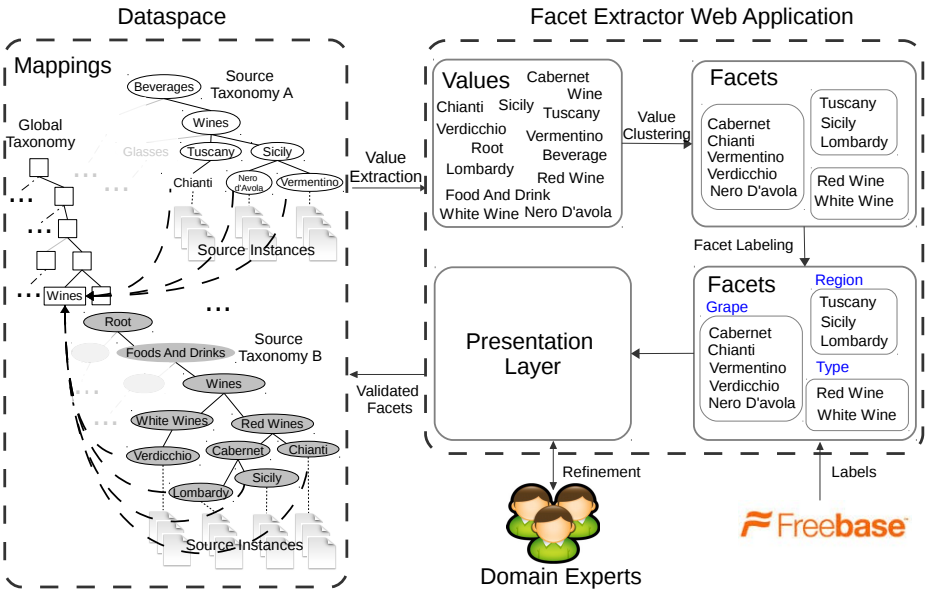

Fig. 2. Overview of the proposed approach to facet extraction

The automatic facet extraction algorithm at the core of the proposed approach is inspired by the following principle: the specialized taxonomies used in data sources contain information that can be analyzed to extract a set of significant facets for a global taxonomy category. The facet extraction algorithm extracts the set of facets $\mathcal{F}^g$ for a global category g using a three-phase process:

1. **Value Extraction:** A set of normalized facet values is produced by case lowerization, special characters removal and stemming of all the source categories mapped to g.
2. **Value Clustering:** Facet values are clustered together into facets according to source taxonomies structural analysis. Since we look for facets of mutual exclusive values, we admit facets containing at least two values. Thus, values that cannot be added to any facet (i.e., clusters of one element) are discarded.
3. **Facet Labeling:** Facets are labelled using external knowledge sources.

The third phase of the facet extraction process is aimed to suggest labels to domain experts' who can accept the suggestion or change the label. We assume that all the taxonomies are lexicalized in a same language. However, our approach does not depend on a particular language: frequency and structural-based principles are used to select and cluster facet values; state-of-the-art Natural Language Processing techniques available in nearly any language are used for facet value normalization; external knowledge sources like the one we used for facet labeling are now available in several languages.

3.1 Value Extraction

During this phase we identify the set of facet values that are frequently used for categorization at the sources. In order to identify such values for a global category g we rely on existing mappings to g. For each source taxonomy S we form the set N_S^g of values occurring as names of source categories mapped to g and all their ancestors in the respective source taxonomy. The level of detail of source taxonomies can be different in each source taxonomy, thus ancestors' names are included in N_S^g to consider every possible significant value. The set N_S^g for a global category g and a source taxonomy S is defined as $N_S^g = \{s \mid \exists\ g \leftarrow s \text{ or } \exists\ g \leftarrow s',\ with\ s \in S\ and\ s'\ is\ a\ descendant\ of\ s\}$.

The set V_S^g of normalized values is obtained by applying case lowerization, special characters removal and stemming to N_S^g. As far as stemming is concerned, we use Hunspell Stemmer[6] to normalize values' terms with respect to their singular form. Hunspell stemmer is based on language dependent stemming rules that are available for most of languages. Normalized values are then unioned together to form the set V^g of facet values for a global category g. In this phase, duplicated values are removed. The set V^g of unique values for a global category g over all the n source taxonomies is $V^g = \bigcup_{i=1}^{n} V_{S^i}^g$.

After normalization and unioning, a simple ranking function is applied to V^g. Unique values are ranked according to their frequency over the all sets $V_{S^i}^g$. Intuitively, the more a value occurs as source category name mapped to the global category g, the higher rank it will get. Based on this ranking we reduce V^g to the set V_k^g of the top k frequent values. The rationale behind this choice is to keep only those values that are more commonly used across many independent and heterogeneous sources and thus are likely to be more relevant for the fine-grained classification of dataspace instances. The set V_k^g of the top frequent values produced by this phase represents the input for the next phase. In addition, we keep track of the (possibly) many source categories to which each value $v \in V_k^g$ correspond. In this way, the annotation of the dataspace instances with facet values extracted by the algorithm is straightforward.

Example. Given the two taxonomies A and B in Figure 2, for the global category "Wines", N_A^{wines} = {"Beverages", "Wines", "Tuscany", "Chianti", "Sicily", "Nero d'Avola", "Vermentino"} and N_B^{wines} = { "Root", "Food And Drinks", "Wines", "White Wines", "Verdicchio", "Red Wines", "Cabernet", "Lombardy", "Sicily", "Chianti"}. After normalization, values from N_A^{wines} and N_B^{wines} form the set of unique values V^{wines} = {"Root", "Beverage", "Food And Drink", "Wine", "Lombardy", "Cabernet", "Tuscany", "Chianti", "Sicily", "Vermentino", "Nero d'Avola", "White Wine", "Verdicchio", "Red Wine"}.

3.2 Value Clustering

Values in V_k^g are clustered to form the set of facets $\mathcal{F}^g$. We aim at clustering together all values that are more likely to be coordinate values of a same

[6] `http://hunspell.sourceforge.net/`

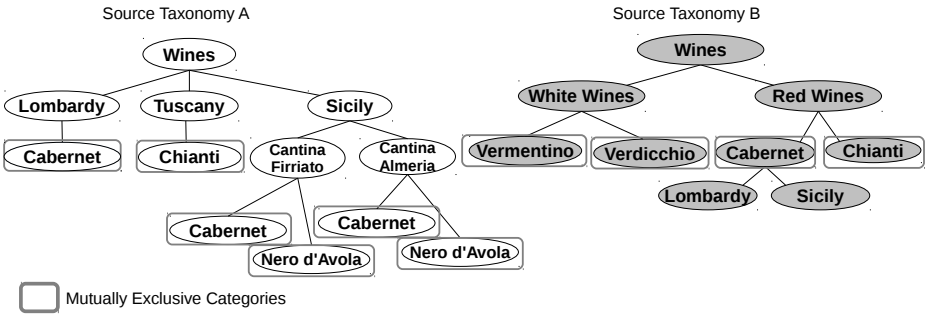

Fig. 3. An example of mutually exclusive source categories

characteristic. As an example, suppose that we get $V_k^{wines} = \{$"Cabernet", "Chianti", "Lombardy", "Sicily"$\}$. An ideal clustering should be $F_1^g = \{$"Cabernet", "Chianti"$\}$ and $F_2^g = \{$"Lombardy", "Sicily"$\}$ because each facet refers to a same characteristic (i.e., the wine's grape variety and the origin Italian region).

In order to discover the set $\mathcal{F}^g$ of facets over V_k^g we make use of the DBSCAN *density-based* clustering algorithm [4]. DBSCAN clusters together values within a maximum distance threshold ϵ and satisfying a cluster density criterion and discards as noise values that are distant from any resulting cluster. DBSCAN algorithm requires in input the minimum cardinality of expected clusters $minPoints$ and the maximum distance threshold ϵ. We set $minPoints$ to 2 and empirically find the best value for ϵ (see Section 4). Finally, DBSCAN does not require a number of expected clusters as input. We use DBSCAN for several reasons. We deal with heterogeneous taxonomies, thus we cannot make any assumption about the shape of clusters (i.e. facets) and we must employ clustering techniques that incorporate the notion of noise. Otherwise, clustering algorithms requiring the expected number of clusters as input are not suitable (e.g. KMeans) since the number of facets to detect is not known in advance.

In order to use the DBSCAN clustering algorithm it is crucial to provide an effective distance metric between the values. We propose a distance metric that considers near those values that refer to a same characteristic of instances, according to a taxonomic structural criterion. We now formally define the proposed distance metric, starting from the principle that it aims at capturing: source categories mutual exclusivity.

Source Category Mutual Exclusivity Principle. We recall from Section 2 that a facet is "a clearly defined, mutually exclusive, and collectively exhaustive aspect, property, or characteristic of a class or specific subject" [19]. The Source Category Mutual Exclusivity principle (SCME) states that the more two values refer to mutually exclusive categories, the more they should be grouped together into the same facet. Given two source categories s_1 and s_2, their occurrence as siblings indicates that s_1 and s_2 are mutually exclusive (e.g., "Lombardy" and "Sicily" in Figure 3). SCME is a *structural* principle: it takes source taxonomies structure into account by considering reciprocal relationships among categories.

Taxonomy Layer Distance. We propose a distance metric that captures the SCME principle by considering sibling relationships between source categories, or more generally, the co-occurrence of categories on a same taxonomy layer. Given a taxonomy S, a taxonomy *layer* l^S of S is the set of all categories that are at the same distance from the taxonomy root. For example, the set {"Lombardy", "Tuscany", "Sicily"} is a layer for taxonomy A in Figure 3. At large scale, categories occurring on same taxonomy layers are likely to be mutually exclusive since they usually represent partitions of the set of entities categorized under the considered taxonomy. Considering co-occurrences on the same taxonomy layer represents a good way to capture the SCME principle: the more two values v_1 and v_2 co-occur at the same layer across all source taxonomies, the more they should be clustered together and thus the less they are distant from each other.

We compute the Taxonomy Layer Distance (TLD) between two values v_1 and v_2 by counting their co-occurrences on the same taxonomy layer and scaling it by their nominal occurrences across all source taxonomies. Computing TLD is equivalent to computing the Jaccard Distance between the two sets of taxonomy layers where two values v_1 and v_2 occur, respectively. Given a value v and a source taxonomy S we define the set L_v^S of layers containing v in S as $L_v^S = \{l^S \mid v \in l^S\}$ (a category can occur in more than one layer). The overall set L_v of layers containing v is computed by unioning all layers across all n source taxonomies as $L_v = \bigcup_{i=1}^{n} L_v^{S^i}$. Then we compute the Jaccard Distance between L_{v_1} and L_{v_2}:

$$\mathrm{TLD}(v_1, v_2) = 1 - \frac{|L_{v_1} \cap L_{v_2}|}{|L_{v_1} \cup L_{v_2}|} \tag{1}$$

Example. Given the two source taxonomies from Figure 3 and values "Cabernet" and "Chianti", we first compute layers containing "Cabernet" and "Chianti":
$l_1^A = \{$"Cabernet", "Chianti", "Cantina Firriato", "Cantina Almeria"$\}$
$l_2^A = \{$"Cabernet", "Nero d'Avola"$\}$
$l_1^B = \{$"Vermentino", "Cabernet", "Verdicchio", "Chianti"$\}$
The set of layers containing "Cabernet" and "Chianti" are:
$L_{Cabernet} = L_{Cabernet}^A \cup L_{Cabernet}^B = \{l_1^A,\ l_2^A,\ l_1^B\}$
$L_{Chianti} = L_{Chianti}^A \cup L_{Chianti}^B = \{l_1^A,\ l_1^B\}$
Then, we compute the distance between "Chianti" and "Cabernet" as:

$$\mathrm{TLD}(\text{"Cabernet"}, \text{"Chianti"}) = 1 - \frac{|L_{Cabernet} \cap L_{Chianti}|}{|L_{Cabernet} \cup L_{Chianti}|} = 1 - \frac{2}{3} = \frac{1}{3}$$

3.3 Facet Labeling

During the labeling phase a semantically meaningful label is attached to each facet F^g discovered in the value clustering phase. Ideally, each label should shortly describe the characteristic to which the values of a facet are likely to refer. For example, if we consider two facets $F_1^g = \{$"Cabernet", "Chianti"$\}$ and $F_2^g = \{$"Italy", "France"$\}$, meaningful labels can be "Grape Variety" and "Country", respectively. Conceptually, labeling each facet means responding to the following question: "to which concept of reality do values of facet F^g refer?".

In order to answer this question, we reconcile each facet value to entities from the Freebase[7] multilingual knowledge base. Given a facet F^g, we submit each facet value of F^g as a keyword query to the Freebase Search Web service[8]. This API performs keyword search over Freebase entities and returns a list of entities ranked by relevance. We select the entity type of the top k ranked entities returned by the keyword search API. We pick as label the most frequent type returned by the Freebase Search API for all facet values in F^g.

4 Evaluation

The core idea of our proposed approach to facet extraction is that we group facet values according to a structural criterion (i.e., TLD). We focus on evaluating the facet value clustering phase. Our goal is to show that TLD effectively captures the SCME principle and supports domain experts in facets definition. To the best of our knowledge there are no distance metrics for taxonomies that explicitly aim at capturing the SCME principle. However, structural similarity metrics that consider path distance between categories within a taxonomy are good candidates to compare our work to. Intuitively, the more two source categories co-occur in the same source taxonomy path (i.e. they are similar to some degree according to structural similarity metrics) from the root to a leaf, the less they are mutually exclusive and the more they should be clustered into different facets (i.e., the clustering algorithm should consider them distant from each other).

We compare TLD with two known structural concept similarity metrics, namely Leacock and Chodorow [9] (LC) and Wu and Palmer [22] (WP) metrics. Both LC and WP achieve high effectiveness results in determining the similarity of concepts within the WordNet taxonomy [16]. LC measures the similarity between two taxonomy categories by considering the shortest path between them and scaling it by the depth of the taxonomy. Similarly, WP measures the similarity between two categories by considering the distance from their nearest common ancestor and the distance of the nearest common ancestor from the taxonomy root. We adapted LC and WP to the case of multiple taxonomies. More specifically, given two values v_1 and v_2 we evaluate their LC and WP similarities for each source taxonomy where v_1 and v_2 co-occur and we take the mean similarity as the final distance value.

4.1 Gold Standard

We created a gold standard from the real world TrovaPrezzi Italian PCE dataspace. We chose ten TrovaPrezzi global categories and ran the Values Extraction phase over them. We presented the set of top k frequent values to TrovaPrezzi domain experts, who found that relevant facet values generally appear among the top 100 ranked values. Thus we choose $k = 100$ as cardinality of the set of

[7] `http://www.freebase.com/`
[8] `https://developers.google.com/freebase/v1/search-overview`

extracted facet values. Facet values were manually grouped together by a domain expert from TrovaPrezzi mapping team and facets were then validated by other domain experts in order to ensure their correctness. As we expected, some of the values were discarded by domain experts as they could be added to any existing facet.

Gold Standards' global categories cover different domains and a relevant portion of the overall dataspace of the PCE, that is 688 source taxonomies and 22594 leaf mappings. For each source taxonomy an average of about 33 mappings have been specified. Moreover, for 322 source taxonomies mappings to more than one global category have been specified. Notice that all the data upon which we created the gold standard are lexicalized in Italian. Our approach to facet extraction is language independent, thus results that we present in following sections are comparable to others obtained considering different languages. For sake of clarity, we provide examples translated to English.

4.2 Evaluation Metrics

We evaluate our facet extraction approach from two different perspectives: facet value effectiveness and value clustering effectiveness. This kind of evaluation campaign has been previously used to evaluate several facet extraction algorithms [8,3]. We introduce the notation we will use in the rest of the section. Given a global category g, we denote with $\mathcal{V}^g$ the set of discovered facet values (i.e. values that have not been classified as noise by the algorithm). We denote with $\mathcal{V}^g_*$ the set of gold standard facet values (i.e., values not classified as noise by domain experts). Lastly, we denote with $\mathcal{F}^g_*$ the set of manually discovered facets (i.e., the gold standard for the global category g), which is compared to the set $\mathcal{F}^g$ of automatically discovered facets.

Value Effectiveness. In our proposed approach noisy values are discarded. In order to evaluate the ability of our technique to filter noisy values out we compare sets $\mathcal{V}^g$ and $\mathcal{V}^g_*$, using Precision (P), Recall (R) and F-Measure (F_1). All these metrics do not take clustering effectiveness into account.

Value Clustering Effectiveness. We evaluate clustering effectiveness using several standard clustering quality metrics, that are Purity (P^*), Normalized Mutual Information (NMI^*), Entropy (E^*), and F-Measure for clustering (F^*). One remark about the usage of these evaluation metrics is that the set of facet values clustered by our approach is different from the set of facet values grouped by humans (i.e., $\mathcal{V}^g \neq \mathcal{V}^g_*$). We may fail in including meaningful values into some clusters, or we may mistakenly include noisy values into some facets. Clustering quality metrics cannot handle these cases. Thus, we modify facets in $\mathcal{F}^g$ by (1) removing all noisy values and by (2) adding to $\mathcal{F}^g$ as single value facets all gold standard values that have been automatically classified as noise. These adjustment ensures that $\mathcal{V}^g = \mathcal{V}^g_*$ and thus clustering quality metrics can be used properly. With this adjustment, facet value effectiveness is not considered.

Table 1. Effectiveness of TLD, LC and WP metrics

	Value Effectiveness			*Clustering Effectiveness*				*Quality*
	P	R	F_1	F^*	NMI^*	Purity	E^*	PRF^*
LC_n	0.359	0.447	0.370	0.403	0.603	**0.308**	**0.243**	0.359
LC_q	0.394	0.953	0.537	0.666	0.709	0.220	0.685	0.531
WP	0.377	**0.984**	0.525	0.682	0.714	0.210	0.744	0.520
TLD	**0.416**	0.901	**0.541**	**0.719**	**0.746**	0.286	0.416	**0.558**

Overall Quality. In order to evaluate the overall effectiveness of our approach, we aggregate facet value precision P, facet value recall R and clustering F-measure F^* into an overall quality measure. The PRF^* measure combines P, R and F^* by means of an armonic mean:

$$PRF^* = \frac{3 * P * R * F^*}{R * P + P * F + P * R} \tag{2}$$

4.3 Experimental Results

We conducted several experiments, comparing clustering performance of TLD, LC and WP metrics. We recall from Section 3.2 that the DBSCAN algorithm used for clustering is configured with a maximum distance threshold ϵ. Optimal values of ϵ depend on the used distance metric, and influence clustering performance. The tuning of ϵ can be driven by two orthogonal factors: overall quality (i.e., PRF^*) and the number of discovered clusters. High values of ϵ (i.e., quality oriented configuration) can lead to better overall quality, but fewer discovered clusters (i.e., the clustering algorithm will tend to group values into one single cluster). Lower values of ϵ (i.e., cluster number oriented configuration) can lead to lower quality, but more discovered clusters. We found that quality oriented and cluster number oriented configurations generally coincide except for LC. In the following section we refer to quality oriented configuration of LC as LC_q while we indicate with LC_n the corresponding cluster number oriented configuration. Since optimal configurations for WP and TLD coincide we omit pedices for them. Moreover, due to space limitation we include only the mean value of metrics computed across all gold standard categories.

Table 1 presents results of our experiments. TLD is more effective in finding relevant facet values and discarding noisy ones, as indicated by an higher F_1. The ability of effectively discarding noisy values substantially reduces domain experts' effort in validating discovered facets. LC_q and WP obtain almost perfect value recall, but substantially lower precision. Thus, they do not effectively support domain experts. Moreover, TLD achieves best performance according to quite all clustering effectiveness metrics, with the exceptions of purity and entropy for LC_n. Clusters discovered by LC_n contain more homogeneous values, in the sense that they have been manually classified as belonging to the same gold standard group. However, LC_n achieves better purity and entropy at the cost of discarding most of the values as noise, thus sacrificing overall quality.

Table 2. Number of groups discovered by TLD, LC, and WP for each source category, compared to the gold standard

	$\lvert\mathcal{F}_*^g\rvert$	LC_q	LC_n	WP	TLD
Dogs and Cats Food	3	1	5	1	7
Grappe, Liquors, Aperitives	1	1	5	1	6
Wines	3	1	1	1	6
Beers	2	6	4	3	14
DVD Movies	2	2	3	1	3
Rings	4	1	6	2	7
Blu-Ray Movies	2	2	3	2	5
Musical Instruments	6	1	3	1	5
Ski and Snowboards	1	1	3	1	7
Necklaces	8	2	6	3	11

The difference between TLD and state-of-the-art metrics is even more evident if we consider the number of detected clusters for each gold standard category g (Table 2). WP and LC_q fail in properly partitioning the overall set V_k^g of facet values, thus failing in detecting groups (i.e. they detect only one or two clusters). They are too inclusive and thus they group facet values at a granularity level that is too high to be suitable for effectively supporting domain experts in bootstrapping a faceted classification system within the dataspace. From the other side, LC_n discards too much values to be effective.

In addition to standard evaluation metrics, we provide a more intuitive insight of results of the facet extraction process, using TLD for facet value clustering compare to state-of-the-art metrics. Table 3 depicts an example of facets discovered for the global category "Wines" by TLD, WP, LC_q, and LC_n compared to manually defined ones. Validating and refining groups discovered by TLD requires much less domain experts' effort than LC_q, LC_n and WP, thus sensibly reducing the cost of bootstrapping faceted classification.

Table 3 highlights a difficulty of TLD in grouping together different lexicalizations of same values (e.g., "Red Wine" and "Red"). One naive approach to overcome this difficulty is to normalize source category names by removing terms belonging to the global category for which the facets are extracted (e.g., the term "Wine" when extracting facets for global category "Wines"). However, this naive solution cannot be generalized to every global category. For example, if we remove from the source category "Dog Food" all the terms belonging to the gold standard global category "Dogs and Cats Food" we end up with an empty, inconsistent facet value. Moreover, also the more conservative approach of removing global category terms only if they *all* occur in the source category cannot be generalized. For example, if we consider the gold standard category "Musical Instruments", using the more conservative approach we will not normalize source categories "Wind Instruments" and "Winds".

We implemented and evaluated both the previously described naive solutions (we omit them due to space limitation), and found that they both decrease the effectiveness of our approach. We believe that effectively solving the problem of

Table 3. Discovered facets for TLD, WP and LC compared to manually discovered facets. Numbers after groups indicate group cardinality.

LC_q	F_1^g = {Wine, Red Wine, White Wine, ..., Piedmont, Lombardy, ..., Sicily, Donnafugata, Cusumano, ..., Alessandro di Camporeale, ..., France} (98)
LC_n	F_1^g = {Wine, Red Wine, White Wine, ..., France, ..., Chianti} (36)
WP	F_1^g = {Wine, Red Wine, White Wine, ..., Piedmont, Lombardy, ..., Sicily, Donnafugata, Cusumano, ..., France} (100)
TLD	F_1^g = { Piedmont, Tuscany, Sicily, , ..., France } (14) F_2^g = { Red, White, Rosé } (3) F_3^g = { Red Wine, White Wine, Rosé Wine } (3) F_4^g = { Moscato, Chardonnay, ..., Merlot } (13) F_5^g = { Tuscany Wine, Sicily Wine} (2) F_6^g = { Donnafugata, Cusumano, ..., Principi di Butera } (27)
Gold Standard	F_1^g = { Piedmont, Lombardy, ..., Sicily } (21) F_2^g = { Red Wine, White Wine, ..., Rosé Wine } (14) F_3^g = { Donnafugata, Cusumano, ..., Alessandro di Camporeale} (12)

different lexicalizations requires Natural Language Processing language specific techniques. NLP techniques can be used to discriminate between global category terms that refer to nouns, verbs, etc. and thus can be safely removed from source categories without creating inconsistencies or change category names' semantics. Introducing this kind of NLP language specific techniques comes at the cost of sacrificing the language independence of our approach. However, this represents an interesting extension of our approach.

5 Related Work

Many different approaches to the problem of extracting facets from structured and unstructured Web resources have been proposed (see [21] for a recent survey). Facets are usually extracted from a *document collection* (e.g., [2,18,13,20]), from search engine *query results* (e.g., [23,8,3,7]) or from the combination of documents and search engine *query logs* (e.g., [15,14,10]).

Document collection based approaches tackle the problem of extracting faceted taxonomies across a document collection. Faceted taxonomies represent a hierarchy topics to which document refer to. External structured resources such as WordNet [2,18], Wikipedia [2,20] or its Linked Data version DBPedia [13] are exploited to enrich the extracted set of facets. Our approach is different from document collection based ones because: (1) we extract facets and facet values in stead of a hierarchy of topics and (2) we analyze taxonomy structure in order to provide sets of mutually exclusive facet values.

The focus of query result based approaches is on classification of documents returned by a keyword query search. Facets for browsing results of a query are extracted from Wikipedia documents [23] analyzing, among other things, Wikipedia categories and reciprocal links between documents. In more general approaches facets are extracted by analyzing raw HTML pages in order to identify potential faceted classifications within them using unsupervised [3] or supervised [8] machine learning techniques. Facets are also extracted by images annotated with a folksonomy [7] and external resources are exploited for

value disambiguation and hyponym detection. State of art query result based approaches deal with the specific problem of integrating and ranking heterogeneous facets that are already present in documents. Our approach takes in input source taxonomies and mappings between them and the global taxonomy.

The focus of query logs approaches is on the usage of user query statistics to identify facet values that are useful/relevant. Query logs are analyzed in order to select relevant facet values with respect to closed, fixed [15] or open, not defined a-priori [14,10] set of facets. Query log based approaches are strictly dependent on end-user queries: they do not consider currently available dataspace instances. Our approach analyzes classifications at the sources, and thus provide a more comprehensive fine-grained dataspace instances classification.

All the previously described approaches are complementary to ours. We extract facets from a different source than previous approaches: taxonomies used to categorize instances within a dataspace. We expect the facet extraction process to benefit from the integration of state-of-the-art facet extraction techniques.

Taxonomy structure analysis has been employed in the field of Ontology Matching [17]. In this field, several similarity metrics between ontology (and also taxonomy) concepts have been proposed and/or adapted from other domains [1,9,22,12]. However, experimental results provided in this paper prove that our proposed distance metric (i.e. TLD) is more effective in capturing the mutual exclusiveness of concepts across multiple heterogeneous taxonomies.

6 Conclusions

In this paper we proposed a semi-automatic, language independent approach to the problem of facets extraction from heterogeneous taxonomies within dataspaces. We proposed a novel metric designed ad-hoc to capture source categories mutual exclusivity across taxonomies. We used the proposed metric as a clustering distance metric for grouping together mutual exclusive facet values. Experimental results show that our approach outperforms state-of-the-art taxonomy concepts similarity metrics in capturing category mutual exclusiveness. Our approach provides valuable aid and reduces domain experts' effort in bootstrapping dataspace fine-grained classifications.

We plan to extend our approach along different directions. Advanced NLP techniques can be used to improve the facet labelling phase and to normalize source categories considering different lexicalizations. Finally, the effective integration of the proposed approach with evidence coming from the consideration of different additional input (e.g., user queries) as proposed in related work is currently under investigation.

References

1. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013, Part II. LNCS, vol. 8219, pp. 294–309. Springer, Heidelberg (2013)

2. Dakka, W., Ipeirotis, P.G.: Automatic extraction of useful facet hierarchies from text databases. In: ICDE, pp. 466–475 (2008)
3. Dou, Z., Hu, S., Luo, Y., Song, R., Wen, J.R.: Finding dimensions for queries. In: CIKM, pp. 1311–1320 (2011)
4. Ester, M., Kriegel, H.P., S, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
5. Halevy, A.Y.: Why your data won't mix. ACM Queue 3(8), 50–58 (2005)
6. Halevy, A.Y., Franklin, M.J., Maier, D.: Principles of dataspace systems. In: PODS (2006)
7. Kawano, Y., Ohshima, H., Tanaka, K.: On-the-fly generation of facets as navigation signs for web objects. In: Lee, S.-G., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 382–396. Springer, Heidelberg (2012)
8. Kong, W., Allan, J.: Extracting query facets from search results. In: SIGIR, pp. 93–102 (2013)
9. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification, pp. 265–283. MIT Press (1998)
10. Li, X., Wang, Y.Y., Acero, A.: Extracting structured information from user queries with semi-supervised conditional random fields. In: SIGIR, pp. 572–579 (2009)
11. Madhavan, J., Cohen, S., Dong, X.L., Halevy, A.Y., Jeffery, S.R., Ko, D., Yu, C.: Web-scale data integration: You can afford to pay as you go. In: CIDR, pp. 342–350 (2007)
12. Mazuel, L., Sabouret, N.: Semantic relatedness measure using object properties in an ontology. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 681–694. Springer, Heidelberg (2008)
13. Medelyan, O., Manion, S., Broekstra, J., Divoli, A., Huang, A.-L., Witten, I.H.: Constructing a focused taxonomy from a document collection. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 367–381. Springer, Heidelberg (2013)
14. Pasca, M., Alfonseca, E.: Web-derived resources for web information retrieval: from conceptual hierarchies to attribute hierarchies. In: SIGIR, pp. 596–603 (2009)
15. Pound, J., Paparizos, S., Tsaparas, P.: Facet discovery for structured web search: a query-log mining approach. In: SIGMOD, pp. 169–180 (2011)
16. Schwartz, H.A., Gomez, F.: Evaluating semantic metrics on tasks of concept similarity. In: FLAIRS (2011)
17. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng. 25(1), 158–176 (2013)
18. Stoica, E., Hearst, M.A., Richardson, M.: Automating creation of hierarchical faceted metadata structures. In: HLT-NAACL, pp. 244–251 (2007)
19. Taylor, A.G., Wynar, B.S.: Wynar's introduction to cataloging and classification. Libraries Unlimited (2004)
20. Wei, B., Liu, J., Ma, J., Zheng, Q., Zhang, W., Feng, B.: Dft-extractor: a system to extract domain-specific faceted taxonomies from wikipedia. In: WWW (Companion Volume), pp. 277–280 (2013)
21. Wei, B., Liu, J., Zheng, Q., Zhang, W., Fu, X., Feng, B.: A survey of faceted search. J. Web Eng. 12(1-2), 41–64 (2013)
22. Wu, Z., Palmer, M.S.: Verb semantics and lexical selection. In: ACL, pp. 133–138 (1994)
23. Yan, N., Li, C., Roy, S.B., Ramegowda, R., Das, G.: Facetedpedia: enabling query-dependent faceted search for wikipedia. In: CIKM, pp. 1927–1928 (2010)