



Automatic Facet Generation and Selection over Knowledge Graphs

Leila Feddoul^{1,2} , Sirko Schindler² , and Frank Löffler¹ 

¹ Heinz Nixdorf Chair for Distributed Information Systems,
Friedrich Schiller University Jena, Jena, Germany
{leila.feddoul, frank.loeffler}@uni-jena.de

² Institute of Data Science, German Aerospace Center DLR, Jena, Germany
{leila.feddoul, sirko.schindler}@dlr.de

Abstract. With the continuous growth of the Linked Data Cloud, adequate methods to efficiently explore semantic data are increasingly required. Faceted browsing is an established technique for exploratory search. Users are given an overview of a collection’s attributes that can be used to progressively refine their filter criteria and delve into the data.

However, manual facet predefinition is often inappropriate for at least three reasons: Firstly, heterogeneous and large scale knowledge graphs offer a huge number of possible facets. Choosing among them may be virtually impossible without algorithmic support. Secondly, knowledge graphs are often constantly changing, hence, predefinitions need to be redone or adapted. Finally, facets are generally applied to only a subset of resources (e.g., search query results). Thus, they have to match this subset and not the knowledge graph as a whole. Precomputing facets for each possible subset is impractical except for very small graphs.

We present our approach for automatic facet generation and selection over knowledge graphs. We propose methods for (1) candidate facet generation and (2) facet ranking, based on metrics that both judge a facet in isolation as well as in relation to others. We integrate those methods in an overall system workflow that also explores indirect facets, before we present the results of an initial evaluation.

Keywords: Faceted browsing · Facet ranking · Knowledge graph · Exploratory search

1 Introduction

A facet is by definition¹ a particular aspect or feature of something. In the present work, this is applied to a set of resources that could be viewed under different aspects. Each aspect is called a *facet* and consists of several categories, *facet values*, which can be used to filter the initial resource set. The number of resources that are associated with a certain facet value is called *value size*.

¹ Oxford Dictionaries: <https://en.oxforddictionaries.com/definition/facet>.

Considering an example, a list of books can be viewed under the aspect of their *genre*. Choosing the facet value *science fiction*, books of this specific *genre* would be selected. The number of selected resources then corresponds to the value size of the facet value *science fiction*. The same list could be viewed under the aspect of their *publication year*, with each sublist containing only books published in one particular year. These two aspects, *genre* and *publication year*, are just two of the many possible facets for books.

To obtain different facets, we assume each resource to have properties assigned, linking them either to other resources (*genre*, with, e.g., a description for itself) or plain literal values (*publication year*). While our method works on any resource set possessing such properties, we use semantic models as rigorous formulation. In particular, we consider knowledge graphs (KGs). They provide significant advantages for the creation of facets: First of all, assuming the resources are drawn from a rich KG, we automatically get a large amount of direct resource information from their properties. The values of those properties may be resources themselves and can be used to generate indirect facets over the initial resource set. For example, an indirect facet for books can be an *author's place of birth*, where *place of birth* is linked to *author*, not to the *book* itself.

However, considering continuously changing and heterogeneous resources, manually predefining facets is often impractical. Using concepts from large KGs, e.g., the Linked Data Cloud, for semantic annotation induces a large number of possible facets. Hence, an automated method has to rank the large number of candidate facets to be able to pick the most suitable ones among them.

Nevertheless, determining the single, best facet is not enough. Users generally expect a list of facets to choose from. Moreover, this list should not be extremely long, and its items should be “useful” both individually and as collection. Were it not for the requirement of usefulness also as collection, simply choosing the top-k highest-ranked facets would be sufficient. However, avoiding facets that are semantically very close to each other is important as well. After their identification, criteria need to be defined to decide which of the candidates to drop to arrive at the final list of facets.

We propose an approach for *dynamic facet generation* and *facet ranking* over KGs. Our ranking is based on intra- and inter-facet metrics to determine the usefulness of a facet, also in the presence of others. A key aspect is exploiting indirect properties to find *better categorizations*. Since inter-facet metrics have not been satisfactorily addressed so far, we present *semantic similarity* as a usefulness criterion.

Based on our previously proposed workflow [1], we integrated all methods into an initial prototypical implementation [2]. While this leverages data from a specific KG, i.e., Wikidata [3], the methods we describe and use are generally applicable without or with only minimal changes to a wide range of KGs. Possible applications include exploratory browsing of a data catalog of semantically annotated datasets, or the reduction of a search result set using facets as filters.

In Sect. 2 we first revisit some of the related works in this direction. We then discuss methods we used for candidate facet generation and ranking in Sect. 3

and propose our workflow in Sect. 4. We present evaluation results in Sect. 5. Finally, we conclude and discuss future work in Sect. 6.

2 Related Work

Faceted browsing over KGs has been the subject of various research efforts, e.g., [4]. Prominent approaches such as *Ontogator* [5] or *mSpace* [6] use *statically predefined* facets for data navigation and do not consider continuously changing data sources. Moreover, their evaluation scenarios suppose data homogeneity and domain-dependent collections like cultural artifacts [5] or classical music [6].

Other projects include *BrowseRDF*, *Parallax*, *gFacet*, *Faceted Wikipedia*, *VisiNav*, *Rhizomer*, *SPARKLIS*, *SemFacet*, *Grafa*, *MediaFaces*, and *Hip-palus* ([7–17], resp.). Facets are either *dynamically selected* from a precomputed set of facets or *dynamically generated* on the fly. The latter type of facets relies on building dynamic SPARQL queries and executing them on the respective SPARQL endpoints. *Grafa* [15] proposed a selection strategy to precompute only a subset of possible facets to avoid indexing of all data.

Some of these projects assume a homogeneous data source [7, 17], using very specific data sets from the domains of, e.g. species [17], other contributions account for *domain heterogeneity* [8–16] and base their work on large scale KGs such as Wikidata [3], Dbpedia [18], or Freebase [19]. However, in some projects [9, 10, 12, 13], an initial interaction (*resource type specification*) is required, before any facets are generated.

Various aspects of facet generation are discussed. This includes *facet ranking* [7, 10–12, 15–17], *entity type pivoting*² [8, 9, 11–14], *visualization* [8, 9, 11–13], *indirect facet generation* [6, 7, 9, 13, 14], or *performance issues* [10, 13, 15].

Facet ranking is of particular importance for dynamic facet generation in order to select from the considerable number of facet candidates. *Frequency-based ranking* was adopted by [10–12, 15]. In *Faceted Wikipedia* [10], facet values are ranked based on the value sizes. For facet ranking, the most frequent facets corresponding to the selected type are candidates. They are ranked based on their most frequent facet value. Note that a ranking is applied only in case of resource type selection, otherwise generic facets are displayed. *VisiNav* [11] also adopts a *frequency-based* approach to rank facets and facet values inspired by PageRank [20]. The respective scores are calculated based on the PageRank score of the data sources [21]. *Rhizomer* [12] defines relevant facets based on the properties usage frequency in the resource type instances and the number of different facet values. In *Grafa* [15], facets are ranked according to the number of search result resources that have a value for the specific facet and facet values are ordered by PageRank. *BrowseRDF* [7] proposes three metrics to measure the quality of facets: (1) predicate balance, considering faceted browsing as the operation of traversing a decision tree where the tree should be well balanced (2) object cardinality, the number of facet values as also considered in [12]

² Switching the focus type, e.g., from a set of *books* to the set of their *authors*.

(3) predicate frequency similar to [10, 12, 15]. The metrics are combined to a final score that is used to rank facets. In *MediaFaces* [16], facets are ranked based on the analysis of image search *query logs* and *users tags* of Flickr³ public images. *Hippalus* [17] introduces a different ranking approach involving user interactions where users rank facets and facet values according to their manually defined *preferences*.

We notice that all the previously described efforts concerning facet ranking only involve *intra-facet metrics* that rate facets individually without taking into consideration the significance of facet co-occurrence, or in other words *inter-facet metrics*. To the best of our knowledge, only *Facetedpedia* [22] includes a metric for measuring the collective usefulness of a facets collection. However, it does not take advantage of KGs or semantically annotated collections, but generates facets over Wikipedia⁴ pages based on the Wikipedia category system. They consider the *navigational cost*, i.e. the number of edges traversed, as an intra-facet metric that is based on the number of steps required to reach target articles and the number of choices at each step. Furthermore, facets are penalized if they have a low *coverage*, i.e., not all the articles can be reached using the considered facet. Besides the *navigational cost*, the *average pairwise similarity* is proposed as an inter-facet metric. However, the used metric is specifically designed to be applied on the Wikipedia category system and is not generic enough to express *semantic similarity* in the sense of arbitrary KGs.

3 Methods

Before presenting our proposed workflow, this section provides details on the employed methods. This includes initial candidate facet generation, handling of literal facet values, and the metrics used to compare facets. The latter discussion is split into two parts: Intra-facet metrics evaluate a facet in isolation, whereas inter-facet metrics judge facets in relation to others.

3.1 Candidate Facet Generation

We aim to generate facets over a set of resources given by their respective Internationalized Resource Identifiers (IRIs) within the KG. In such a graph we treat the relations of the given resources as their properties and thus any applicable property path is equivalent to a candidate facet. To achieve a better categorization of resources, we consider not only the direct properties (i.e., values that are connected to the resource by a single link), but also indirect properties (i.e., chained links are needed to connect a resource and a value). As an example, consider a set of resources referring to people. A direct property can be derived from a relation *place of birth* pointing to instances of a class *city*. An indirect property could then also exploit an existing link between *city* and *country*⁵ to

³ <https://www.flickr.com/>.

⁴ <https://www.wikipedia.org/>.

⁵ Assuming there is no direct link between persons and their country of birth.

arrange the connected cities into possibly fewer categories⁶. Indirect properties are only possible, if the range of the associated relation is not a literal, as those can not be the subject of further statements in the standard RDF model.

A candidate facet is now given by a property path within the KG. In case of direct properties this path is of length one, whereas for indirect properties any path length greater than one is possible. However, longer paths loosen the connection between resources and facets values. At some point this renders a facet useless for the given task or at least makes it unclear to users how that facet is supposed to support them. Furthermore, longer paths increase the number of candidates and thus require more computations in later phases. For these reasons, we limit the path length for candidates by a threshold τ .

We categorize candidate facets into two types: (1) *Categorical facets* that result from property paths connecting exclusively to other resources and (2) *quantitative facets* whose values are given by literals. While we allow quantitative candidates for numeric or date literals, we exclude string literals. The rationale is that those oftentimes contain labels or descriptions specific to single resources and, hence, are barely shared between different ones. As facets rely on common values to categorize the given input set, these properties will only rarely provide a suitable candidate facet. If a string value is common to multiple resources, there is a high chance, that this should have been modeled as a distinct resource instead of a literal. Of course, resources are often not modeled perfectly. Future work might need to include these to be able to cope with this type of data.

3.2 Clustering of Quantitative Facets

As mentioned before, facets can be created from numeric or date literals. Unlike categorical facets, it is highly unlikely that the number of distinct values is sufficiently small to generate a useful facet. However, these values can be clustered by dividing their continuous range into discrete subranges.

The clustering step is only applied to quantitative facets. It replaces the associated values with value ranges. The number of these clusters is determined by the optimum value cardinality as defined by the respective intra-facet metric (see Subsect. 3.3). The clustering technique itself is a consequence of the rationale behind another intra-facet metric, the value dispersion. It assembles approximately the same number of values in each cluster.

3.3 Intra-Facet Metrics

To select the most useful facets among the candidates, we define metrics to judge their usefulness. The first set of metrics presented here assigns scores to individual candidates independently of each other. Each metric is designed to reflect one intuition of what constitutes a useful facet.

The first requirement concerns the applicability of the facet. For each facet we also include an *unknown* value. This accumulates the resources that do not

⁶ Cities belonging to the same country will be grouped into one category.

support the respective property path, i.e., at least one of the corresponding relations is missing for this resource. For heterogeneous resource sets, the *unknown* value size will be non-zero for most facets. However, for a facet to be useful, it should apply to as many resources as possible. So we strive for the value size of *unknown* to be small in comparison with the overall size of the resource set.

These thoughts lead to the definition of *predicate probability* of a facet f , $score_{predicateProb}$, as given in Eq. 1. It calculates, for a randomly chosen resource, the probability to support the property path of a given facet.

$$score_{predicateProb}(f) = \frac{|supporting\ resources|}{|resources|} \quad (1)$$

Our next requirement deals with the number of facet values. We consider a facet with only a single value as not useful, as it can not be used to narrow down the given set of resources. But then again, facets with too many values provide little help as well. Here, users have to scan through a long list of possible options, which may even rival the number of input resources. We believe that there is a number of values that is optimal in the sense that it balances between a concise categorization and a sufficient number of options to choose from.

Following these considerations, we define the *value cardinality*, $score_{valueCard}$, of a facet f with a number of values c_f as given in Eq. 2. The minimum cardinality is denoted by $minCard$ and the optimal one by $optCard$. Note that we chose an asymmetric function that favors facets with fewer values rather than more. This follows the intuition that better categorizations tend to have fewer categories. The parameter $\theta \neq 0$ allows to adjust the preference for value sizes between $minCard$ and $optCard$.

$$score_{valueCard}(f) = \begin{cases} 0 & \text{if } c_f < minCard \\ e^{\frac{c_f - optCard}{\theta^2}} & \text{if } minCard \leq c_f \leq optCard \\ \frac{1}{1 + (c_f - optCard)} & \text{if } c_f > optCard \end{cases} \quad (2)$$

Our final requirement follows the principle of self-balancing search trees: Each decision made while traversing the tree should eliminate roughly the same number of results from consideration. In other words, no leaf node (representing a specific result) is preferred over others in terms of steps needed to reach it from the root node. Similarly, we do not want to favor any specific category.

For a facet, this means that all value sizes within a single facet should be approximately equal⁷. As a measure for the variance in value sizes, we employ the coefficient of variation c_v (see Eq. 3). We chose this coefficient over the plain standard deviation, as it allows to better compare across multiple facets with possibly different value sizes. Using this, we define the *value dispersion*, $score_{dispersion}$,

⁷ The subsets induced by the different facet values do not have to be disjoint. A single resource may be linked to several such values. Consider, e.g., the relation *part of* that relates *country* and *continent*. Here, the individual *Russia* is connected to two continents, *Asia* and *Europe*, thus appearing as part of both facet values' results.

as given in Eq. 4. Here, N is the number of facet values, x_i denotes the value size of the i th facet value, and \bar{x} is the average of all value sizes. We exclude the value size of the special facet value *unknown* from this calculation, as this value is already exploited in *score_{predicateProb}*.

$$c_v(f) = \frac{\sqrt{\frac{1}{N} \times \sum_{i=1}^N (x_i - \bar{x})^2}}{\bar{x}} \quad (3)$$

$$score_{dispersion}(f) = \frac{1}{1 + c_v(f)} \quad (4)$$

All presented metrics are designed to return only values in the range between zero and one. In order to combine them into a single metric used in the ranking process (see Sect. 4), we can use a weighted average as shown in Eq. 5. With the individual weights summing up to one as well, we assure that the final *score* is also between zero and one.

$$\begin{aligned} score(f) = & w_{predicateProb} \times score_{predicateProb} \\ & + w_{dispersion} \times score_{dispersion} \\ & + w_{valueCard} \times score_{valueCard} \end{aligned} \quad (5)$$

$$\text{with } \sum_i w_i = 1$$

3.4 Inter-Facet Metrics

In contrast to their intra-facet counterparts, inter-facet metrics assess the relationship between different candidate facets. We use *semantic similarity* of facets as an inter-facet metric. The motivation is to prevent facets that are too close to one another and thus would provide about the same partitioning of the resource set. Moreover, semantically distant facets increase the chances of meeting users' information need and/or mindset.

Generally, no restrictions are imposed on the *semantic similarity* measure chosen to be included in the current facet generation workflow. However, we base our workflow on a structure-based measure that combines the shortest path length and the depth. In particular, we consider the one proposed by [23] as reference similarity metric between two concepts c_i and c_j , defined as follows:

$$sim(c_i, c_j) = e^{-\alpha \cdot length(c_i, c_j)} \cdot \frac{e^{\beta \cdot depth(c_{lcs})} - e^{-\beta \cdot depth(c_{lcs})}}{e^{\beta \cdot depth(c_{lcs})} + e^{-\beta \cdot depth(c_{lcs})}} \quad (6)$$

where $length(c_i, c_j)$ is the shortest path length between c_i and c_j and $depth(c_{lcs})$ is the shortest path length between the Least Common Subsumer (LCS) of the two concepts, c_{lcs} and the root concept. $\alpha \geq 0$ and $\beta > 0$ are used to adjust the importance assigned to the shortest path length and the depth, respectively. Based on the correlation evaluation conducted by [23], the optimal parameters are $\alpha = 0.2$ and $\beta = 0.6$.

The previously defined *semantic similarity* metric takes a pair of concepts as input. Therefore, a mapping between properties and concepts needs to be available. For this purpose, we exploit a particular characteristic of Wikidata’s data model: Properties are annotated with a matching entity. For example, the property *author* ($P50$) is itself linked to the entity *author* ($Q482980$). This allows us to retrieve entities corresponding to the property path of a facet.

When comparing two facets, we first retrieve the respective entities for the first property in their property paths. We then calculate the *semantic similarity* between the entity pair. Two entities are considered similar, if *sim* is larger than a defined threshold σ . Since we calculate the similarity over Wikidata taxonomy, we only consider links using *subclass of* ($P279$) and *instance of* ($P31$) here.

4 Workflow

We consider the facet generation to be part of larger applications. In particular, we assume that the retrieval of an initial resource set is subject to other independent components. Hence, details of the resource retrieval process are out of scope at this point. For the sake of argument, we base our workflow on the results of a keyword-based full text search over the string properties of entities in the KG. Its result is represented as a set of IRIs, each identifies a single result item or resource and forms the input to our proposed facet generation workflow. We structured the overall process into four phases as shown in Fig. 1.

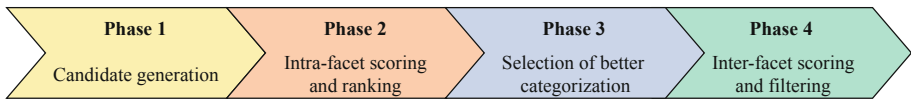


Fig. 1. Phases of the facet generation process.

Phase 1: Candidate Generation

This first phase enumerates possible facets by querying for a list of property-paths associated with the input list of resources. As the predicate probability $score_{predicateProb}$ is a simple metric, we choose to include it as part of the query. Candidates that have a $score_{predicateProb}$ below a predefined threshold, $minPredProb$, are already removed in this phase. This reduces the necessary data transfers and the calculation of computationally expensive metrics. The result is a list of candidates, each comprised of a basic graph pattern (BGP), that describes the facet, and a score to reflect the fraction of resources it applies to.

Phase 2: Intra-Facet Scoring and Ranking

As a prerequisite for the remaining intra-facet metrics, now the facet values along with the respective value size are retrieved from the SPARQL endpoint.

We distinguish between object and data properties⁸ at this point. The latter are subjected to the clustering described in Subsect. 3.2 to derive comparable characteristics with regard to intra-facet metrics.

After augmenting the facets with their respective values, the remaining intra-facet metrics, $score_{dispersion}$ and $score_{valueCard}$, are calculated for all candidates. This allows us to compute the final intra-facet score, $score(f)$, and accordingly rank all facets in decreasing order.

Phase 3: Selection of Better Categorization

The number of necessary inter-facet metrics calculations grows quadratically with the remaining number of candidates. To reduce the list of candidates before the next step, we exploit a key characteristic of the semantic similarity metric. The similarity only depends on the first direct property of each facet. Consequently, out of all candidates sharing the direct property, only one will be chosen for the final result, as all others will be too similar to it. Leveraging this observation, we can group the candidates by their direct properties and only choose the best-ranked one within each group.

Phase 4: Inter-Facet Scoring and Filtering

The final result is derived by consecutively applying inter-facet metrics to chosen pairs of candidates. Calculating semantic similarities is rather expensive. To minimize the comparisons required, facets are selected in a greedy fashion.

Let C be the list of candidates in decreasing order w.r.t. the intra-facet metric scoring of Phase 2 and S be the final collection of facets as returned by Phase 4.

- (i) Initialize S with the best-ranked facet.
- (ii) Take the next facet out of C and compare it with the facets in S .
- (iii) If it is not closely semantically similar to any facet in S , add it to S .
- (iv) Continue with Step (ii) until the desired number of facets is reached or there are no more candidates left.

Finally, S will contain a subset of facets deemed most suitable for the given input set of resources. The suitability has been determined by employing both the intra- and inter-facet metrics, which can be extended or changed without affecting the corresponding workflow. S can now be presented to users. Note that selecting specific value and subsequently reducing the result set will trigger a new facet generation process, as the basis for our calculations—the input resource set—might have changed substantially.

5 Evaluation

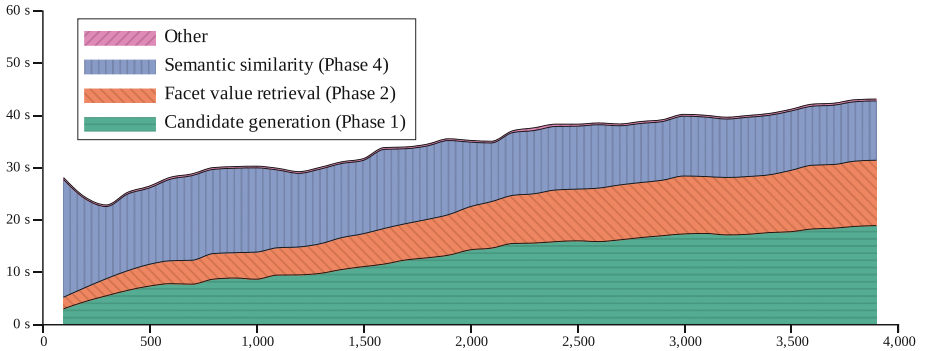
The methods described in Sect. 3 were implemented in a prototype that issues dynamic SPARQL queries to the public SPARQL endpoint of Wikidata (WDQS)⁹. The source code is available online [2], under an MIT license.

⁸ Data properties using string literals have already been excluded in the candidate generation. That means, only numeric and date literals are considered here.

⁹ <https://query.wikidata.org/>.

Table 1. Number of candidates depending on path length and number of IRIs.

#IRIs	100	1000	2000	3000	4000
$\tau = 1$	37	52	65	66	75
$\tau = 2$	901	1643	2039	2342	2648
$\tau = 3$	16076	31543	39318	44619	50843

**Fig. 2.** Benchmark results: average timings depending on the input IRI size.

5.1 Benchmarking

To evaluate the performance of our prototype we used a collection of IRIs extracted from Wikidata (instances of *novel* (Q8261) or its subclasses).

First, we examined the change in the number of candidates depending on the path length τ and number of input IRIs. Results are shown in Table 1. As expected, the number of candidates increases significantly –about 20-fold– for each additional hop in the paths. However, a growth in input IRIs yields only a small effect in comparison. These figures and the considerations of Subsect. 3.1, led to a path length of $\tau = 2$ for the remainder of the evaluation.

Subsequently, we looked at the run-time of our prototype for varying sizes of input IRIs. We fixed the semantic similarity threshold ($\sigma = 0.70$), the parameters for value cardinality scoring ($optCard = 10$, $minCard = 2$, and $\theta = 3$), and the predicate probability threshold ($minPredProb = 0.1$). Figure 2 shows a breakdown of the measured execution times, averaged over about 350 individual measurements over the course of a week. We observe a less than linear growth of run-time depending on the input IRI size. The most expensive operations are (1) *candidate generation*, (2) *facet value retrieval*, and (3) *semantic similarity*. Other operations such as *intra-facet metric* calculation and *selection of better categorization* do not contribute significantly. A detailed analysis revealed that the execution times are largely dominated by querying the SPARQL endpoint.

Overall, we acknowledge that the current performance prohibits any productive use. However, the overwhelming impact of query response times on the

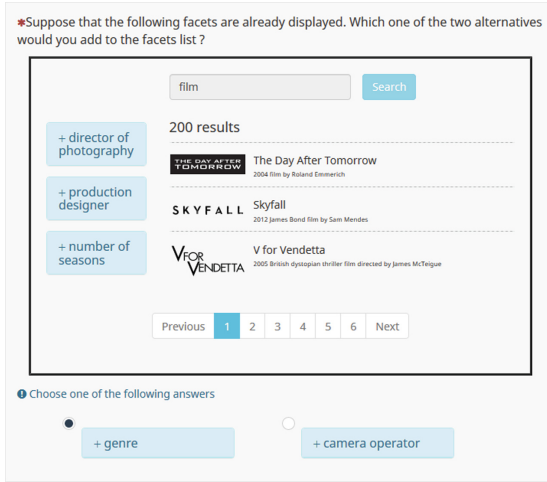


Fig. 3. User evaluation: Fictitious interface for *facet selection* task.

overall execution time indicates potential for improvement. Further parallelization and caching of reoccurring queries might prove fruitful.

5.2 User Evaluation

Setup. In a survey-based user evaluation, we examined whether facets generated by the proposed workflow match user expectations. Based on a fictitious scenario, we assumed an initial search with the keyword “film”.

After introducing users to the general concepts of faceted search and the given scenario, we asked for user preferences in a series of questions categorized into two kinds of situations: one for *facet selection* and one for *facet ranking*. In *facet selection* (cf. Fig. 3), users were presented with a static user interface that resembles a common search engine and includes three different facets, e.g., *director of photography*, *production designer*, and *number of seasons*. They were then given two more facets, e.g., *genre* and *camera operator*, and were asked which would be a better addition to the existing three facets. In *facet ranking*, we presented three to four different facets per question and asked users to rate their usefulness in the given scenario using a five point Likert scale [24].

Unlike *facet selection*, where only facet headers are shown, *facet ranking* also includes facet values. Unless noted otherwise, all facets and their values are modeled according to the data present in Wikidata as of February 2019 using a path length of $\tau = 2$. The facets are generated by an initial, prototypical implementation of the workflow, but were manually adapted to reflect the respective evaluation intent to emphasize specific intra-facet scores.

Using these situations, the following order of questions was used in the survey. Overall, we created a pool of 43 questions, out of which a random subset of 15

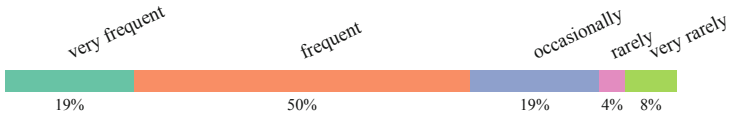


Fig. 4. Usage of facets. An option “never” was provided, but not chosen by any user.

was chosen for each user. This approach is intended to reduce the bias that might arise from certain terms used throughout.

In a first set of questions we focus on inter-facet comparisons using *facet selection*. In particular, this evolves around the *selection of better categorization* (Phase 3 in Sect. 4) and *semantic similarity* (Subsect. 3.4).

A second set of questions uses *facet ranking* with facets modeled after Wikidata. This compares multiple indirect facets with their respective direct counterparts. Here, the indirect facets also vary in their intra-facet scores, allowing us to evaluate our strategy in the *selection of better categorization*.

Finally, we used *facet ranking*, this time with abstract facets, i.e., replacing facet headers with “Facet 1” etc. and values with “Value 1” etc. The reason is again to reduce bias stemming from the actual semantics of the proposed facets. In this last part of the evaluation, we issued questions, where the proposed facets differed only with respect to one intra-facet metric¹⁰. In a similar fashion, we also examined combinations of two and all three proposed intra-facet metrics.

For the survey, we recruited 26 volunteers differing in age (18–44) and educational background. In total, they performed 130 facet selections and 936 individual facet ratings. Most of the participants stated at least an occasional use of facets, if they are provided (cf. Fig. 4). Consequently, we assume that they are familiar with the general behavior of faceted browsing.

Results. For each question in *facet selection*, we derive the percentage of participant selections that match the system decision. Figure 5 shows the results of the first question set with each dot representing agreement of one particular question¹¹. For the *selection of better categorization* we see an overall agreement between the survey users and our system of $\sim 83\%$.

The average result for *semantic similarity* is mixed ($\sim 63\%$). However, when analyzing the agreement per question, we see a more polarized result. While users most often agree on a specific facet, our system is not always able to concur with this choice. This leads us to believe that the survey responses were driven more by the applicability of the individual facet and not its relation to the already given ones. Yet, this is dependent on the available information and hence, out of control of the proposed workflow.

In *facet ranking*, we are not interested in the specific numerical values each metric provides, but focus on the ranking induced by those metrics. To compare the ranking determined by our system with the ranking induced by the sur-

¹⁰ The respective other metrics did not vary within a small error margin.

¹¹ By experiment design, not all questions received the same number of responses.

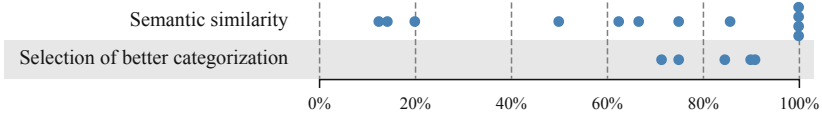


Fig. 5. Agreement of participants and system in *facet selection*. One dot per question.

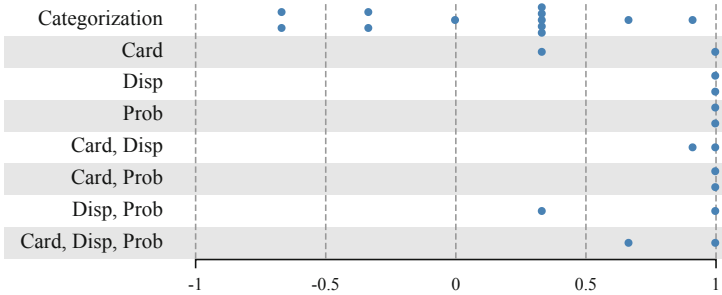


Fig. 6. Rank correlation for *facet ranking* tasks. One dot per survey question. Value Cardinality (*Card*), Value Dispersion (*Disp*), Predicate Probability (*Prob*).

vey responses, we encoded the latter using numerical values and calculated an average rating for each facet. For each question, we ranked the presented facets according to these ratings, which results in a *survey ranking*. We then chose Kendall’s Tau-B¹² to compare our *system ranking* with this *survey ranking*.

The survey responses for the second question set, concerned with the selection of better categorization, are shown in the topmost lane of Fig. 6. The overall result shows no clear support for our approach in this step. When there was no (obvious) relation between the indirect property and the initial resource set (e.g., a facet for *country of origin/driving side*), users rated the facet rather low. However, the system sometimes favors these facets, as they oftentimes provide a good categorization with respect to the defined metrics. On other occasions, like the facet *country of origin/continent*, both users and the system agree that this is a helpful facet. This leads us to believe that, although indirect facets are promising, they require additional refinement to ensure their relevancy.

The final question set verified our metrics independent of semantic biases induced by real-world facets. Results are shown in the lower parts of Fig. 6. In general, survey participants agree almost completely with our approach. The only exceptions are due to a tie (*Card, Disp*) or a different opinion about the order of one particular pair of facets (*Disp, Prob* and *Card, Disp, Prob*).

The user evaluation suggests that the technical criteria seem well suited in isolation. However, resulting facets not only have to be evaluated against each

¹² Kendall’s Tau-B is a variant of Kendall’s Tau that also accounts for possible ties in the ranking. Values range from +1 for identical rankings to −1 for inverse ones. A value of 0 hints towards no correlation between the involved rankings.

other, but also against the semantic context of the input IRIs. While in search tasks user input can be used to assess this intent, it remains open how this can automatically be approximated for arbitrary resource sets.

6 Conclusion

We have proposed methods to enable automatic facet generation and ranking over KGs. In particular, we provided an approach for dynamic candidate facet generation for arbitrary input sets of resources. We defined intra- and inter-facet metrics to rank the candidates and reduce the possible facet space by selecting the most useful ones. We explored indirect properties to find better categorizations and consequently enhance facets' usefulness. We proposed semantic similarity as a criterion to select among multiple candidate facets. Finally, we developed a holistic workflow that integrates all proposed methods.

Initial survey results support the used metrics. While indirect facets show promise as a helpful addition, their relevancy for the initial resource set needs to be ensured. This latter issue is also the main focus of our future efforts: How can we estimate the relatedness to the initial input for indirect facets? Another prime direction is a performance improvement of our initial prototype, to make it applicable for real-world systems (e.g., caching and parallelization of queries).

References

1. Feddoul, L., Schindler, S., Löffler, F.: Semantic relatedness as an inter-facet metric for facet selection over knowledge graphs. In: *ESWC 2019* (2019, in press)
2. Schindler, S., Feddoul, L.: *Semantic Facets* (2019). <https://doi.org/10.5281/zenodo.2784142>
3. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. *Commun. ACM* (2014). <https://doi.org/10.1145/2629489>
4. Tzitzikas, Y., Manolis, N., Papadakos, P.: Faceted exploration of RDF/S Datasets: a survey. *J. Intell. Inf. Syst.* **48**(2), 329–364 (2016). <https://doi.org/10.1007/s10844-016-0413-8>
5. Mäkelä, E., Hyvönen, E., Saarela, S.: Ontogator — a semantic view-based search engine service for web applications. In: Cruz, I., et al. (eds.) *ISWC 2006*. LNCS, vol. 4273, pp. 847–860. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_61
6. Schraefel, M.C., Smith, D.A., Owens, A., Russell, A., Harris, C., Wilson, M.: The evolving mSpace platform: leveraging the semantic web on the trail of the Memex. In: *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia, HYPERTEXT 2005*, pp. 174–183. ACM (2005). <https://doi.org/10.1145/1083356.1083391>
7. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: Cruz, I., et al. (eds.) *ISWC 2006*. LNCS, vol. 4273, pp. 559–572. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_40
8. Huynh, D., Karger, D.: *Parallax and companion: set-based browsing for the data web*. Technical report, Metaweb Technologies Inc. (2009)

9. Heim, P., Ziegler, J., Lohmann, S.: gFacet: a browser for the web of data. In: Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW 2008), Aachen (2008)
10. Hahn, R., et al.: Faceted wikipedia search. In: Abramowicz, W., Tolksdorf, R. (eds.) BIS 2010. LNBI, vol. 47, pp. 1–11. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12814-1_1
11. Harth, A.: VisiNav: a system for visual search and navigation on web data. *J. Web Semant.* **8**(4), 348–354 (2010). <https://doi.org/10.1016/j.websem.2010.08.001>. Semantic Web Challenge 2009 – User Interaction in Semantic Web research
12. Brunetti, J.M., García, R., Auer, S.: From overview to facets and pivoting for interactive exploration of semantic web data. *Int. J. Semant. Web Inf. Syst.* **9**(1), 1–20 (2013). <https://doi.org/10.4018/jswis.2013010101>
13. Ferré, S.: Expressive and scalable query-based faceted search over SPARQL endpoints. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 438–453. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_28
14. Arenas, M., Grau, B.C., Kharlamov, E., Marciuska, S., Zheleznyakov, D.: Faceted search Over RDF-based knowledge graphs. *Web Semant. Sci., Serv. Agents World Wide Web* **37**(0) (2016). <https://doi.org/10.1016/j.websem.2015.12.002>
15. Moreno-Vega, J., Hogan, A.: GraFa: scalable faceted browsing for RDF graphs. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11136, pp. 301–317. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_18
16. van Zwol, R., et al.: Faceted exploration of image search results. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 961–970. ACM (2010). <https://doi.org/10.1145/1772690.1772788>
17. Papadakos, P., Tzitzikas, Y.: Hippalus: preference-enriched faceted exploration. In: EDBT/ICDT Workshops. CEUR Workshop Proceedings, vol. 1133, pp. 167–172. CEUR-WS.org (2014)
18. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web J.* **6**(2), 167–195 (2015). <https://doi.org/10.3233/SW-140134>
19. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 1247–1250. ACM, New York (2008). <https://doi.org/10.1145/1376616.1376746>
20. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, pp. 161–172 (1998)
21. Harth, A., Kinsella, S., Decker, S.: Using naming authority to rank data and ontologies for web search. In: Bernstein, A., et al. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 277–292. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04930-9_18
22. Li, C., Yan, N., Roy, S.B., Lisham, L., Das, G.: Facetedpedia: dynamic generation of query-dependent faceted interfaces for Wikipedia. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 651–660. ACM, New York (2010). <https://doi.org/10.1145/1772690.1772757>
23. Li, Y., Bandar, Z.A., Mclean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.* **15**(4), 871–882 (2003). <https://doi.org/10.1109/TKDE.2003.1209005>
24. Likert, R.: A technique for the measurement of attitudes. *Arch. Psychol.* **22**, 5–55 (1932)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

