Faceted Wikipedia

ABSTRACT

This paper proposes FacetedPedia, a faceted retrieval system for information discovery and exploration over Wikipedia. Given the set of Wikipedia articles resulting from a keyword search query, FacetedPedia dynamically and automatically discovers a faceted interface for navigating and exploring the result articles. The interface consists of multiple facets, with a hierarchy of categories on each facet. Given the sheer size and complexity of Wikipedia, the space of possible choices of faceted interfaces is prohibitively large. We propose metrics for ranking individual facet hierarchies by users' navigational costs in reaching the target articles, and metrics for ranking interfaces (each with k facets) by both their average pairwise similarities and average navigational costs. We design facets discovery algorithms to generate faceted interfaces for the target articles, optimizing the ranking metrics. Our experimental evaluation and user study verify the effectiveness of our methods in generating useful faceted interfaces.

To the best of our knowledge, FacetedPedia is the first faceted retrieval system for Wikipedia. Compared with such systems over multimedia objects, text documents, and relational records, FacetedPedia is fully automatic and dynamic in both facets discovery and hierarchy construction, and the facets are based on rich semantic information in articles. The essence of our approach is to build upon the intensive structures (hyperlinks) and folksonomy (category system) created by collaborative Wikipedia authors. In comparison, existing systems fall into two extreme ends. Some create facets based on pre-defined explicit structures (*e.g.*, relational schema) and semantic information (*e.g.*, domain-specific taxonomies), while other systems do not have enough structures and semantic information that can be leveraged, thus generate facets according to simple IS-A or subsumption relationships between textual terms.

1. INTRODUCTION

Wikipedia has gained enormous popularity since its birth. It is among the top 10 most popular Websites in terms of user traffic [3]. With the 2.5 million English articles by 2008, it has become the largest encyclopedia ever created [2]. The prevalent manner in which the Web users access Wikipedia articles is keyword search, through either general search engines or the search interface of Wikipedia itself. Although keyword search has been quite effective in finding specific target Web pages, we often encounter more sophisticated information discovery and exploratory tasks that call for alternative or complementary access apparatus. Such tasks become even more typical on Wikipedia due to its rich informative articles.

One access mechanism that is potentially useful on Wikipedia is the *faceted interface*, or the so-called *hierarchical faceted categories* (HFC) [14]. The concept is very simple. A faceted interface for exploring and browsing a set of objects is a set of category hierarchies, where each hierarchy corresponds to an individual *facet* (or dimension, or attribute) of the objects. The set of objects can be reached from a facet by navigating through the hierarchy of categories, until reaching the attribute values associated with the objects. Users navigate the multiple facets simultaneously and the intersection of the chosen objects on individual facets are brought to users' attention. The navigation on a faceted interface therefore corresponds to repeated constructions of conjunctive queries with selection conditions on multiple dimensions.

Faceted interfaces, together with clustering [9, 28, 15], are two major competing ideas for user interfaces that organize and group retrieval results. The utility of such user interfaces have been investigated in various studies including [17, 14]. It is shown that users engaged in exploratory tasks often prefer such result groupings over simple ranked result-lists (commonly provided by search engines) which is more appropriate when searching for specific information items [18, 27, 15]. Furthermore, usability studies show that in many situations users do not like the disorderly groupings produced by clustering methods, preferring the more understandable category hierarchies of faceted interfaces [18, 19, 14]. Faceted interfaces have recently gained much popularity and have been applied in multimedia retrieval [27], text [14, 22, 11, 10], relational data [21, 13], metadata-rich text corpus [12, 7], E-commerce vendors (such as Endeca [1], IBM, and Mercado [2]), as well as several E-commerce Websites (*e.g.*, eBay.com).

1.1 Motivations

In this paper we propose Facetedpedia, a faceted retrieval system over Wikipedia. The distinguishing features of Wikipedia articles that separate them from common Web pages make such a system both needed and possible. First, when looking for information on Wikipedia, our interests are not necessarily limited to finding specific target articles. Instead, we often find ourselves performing information discovery on certain subjects by sift-

[1] http://www.endeca.com/
[2] http://www.mercado.com/

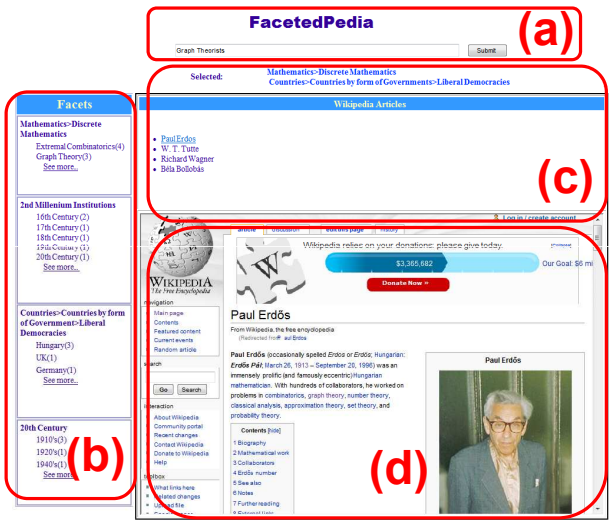


Figure 1: The graphical user interface based on facets.

ing through a lot of relevant articles. Search engines have been highly effective in assisting us find specific pages, but are limited in supporting such discovery process. Therefore access mechanisms enabling exploration and browsing of `Wikipedia` articles, such as faceted interface, would be useful. Second, different from most other Web pages, `Wikipedia` articles contain rich structures and semantic information that are created by users collaboratively. To name just a few, examples of such structures include the categories of articles, intensive internal links between articles, explicit metadata information (*e.g.*, so-called *infoboxes*), lists and tables, and sectioning of individual articles. The rich structures provide the substance to enable faceted retrieval.

We give below a motivating example of `Facetedpedia`. The hypothetical faceted interface in the example could be useful in many places, since we may encounter similar information discovery tasks in other scenarios.

Example 1 (Motivating Example): Imagine that a graduate student, Amy, doing her PhD in Theoretical Computer Science, is exploring different information about renowned graph theoreticians. Impressed by the rich information content and popularity of `Wikipedia`, she decides to explore its relevant articles. The current tools available for her exploration are either a search engine or the search interface of `Wikipedia` to which she can issue a keyword query such as {"Graph", "Theorists"}. However, as it has been argued earlier, search engines in themselves are unsuitable for such exploratory tasks. Alternatively, Amy can try browsing the category system available in `Wikipedia`, where she can start from the root and follow relevant categories and sub-categories to reach various articles of interest. However, the vast size and complexity of the category system would force Amy to evaluate each each category/subcategory for relevance at every step, making the exploration task exhaustive and impractical.

In contrast, a faceted interface where multiple facets are automatically and dynamically derived to cover the result articles (e.g., the top 200 pages retrieved by a search engine in response to Amy's initial keyword query), with each facet describing a different "dimension" of the results, would be very helpful to Amy. For example, the following facets could be useful in navigating and exploring articles related to Graph Theoreticians.

`Mathematics>Discrete Mathematics`
`Countries>Countries by form of Government>Liberal Democracies`
`20th Century>1910's`

Under this faceted interface, each article can be assigned to many nodes in these hierarchies, with each assignment representing an attribute value of the article. For example, the `Wikipedia` article on graph theoretician Paul Erdos can be described by the following attribute values.

`Mathematics>Discrete Mathematics>Graph Theory`
`Countries>Countries by form of Government>Liberal Democracies>Hungary`
`20th Century>1910's>1913`

Using such a faceted interface, Amy can simultaneously navigate down the multiple facets to easily explore the covered articles.

The interface of `FacetedPedia` is shown in Figure 1. The system takes a keyword search query from the user as the input (region (a) in Figure 1) and obtains a ranked list of search result articles. The system automatically generates k facets (region (b) in Figure 1) for the top s articles in this ranked list. On each facet, the current category path that the user has selected is shown on the top, followed by the available subcategories following the current category path, allowing easy navigation by the user. The interface also shows the titles of the reachable target articles (region (c) in Figure 1). When the user clicks one title, the corresponding `Wikipedia` article would be shown (region (d) in Figure 1). When the user selects an attribute article, she reaches the end on that facet. ■

1.2 Overview of Challenges and Solutions

In `Facetedpedia` we focus on the problem of *automatic* and *dynamic* discovery of faceted interfaces such as the one in Example 1. That is, given the set of top-s ranked `Wikipedia` articles as the result of a keyword search query, `Facetedpedia` produces a query-dependent interface of multiple facets for exploring the result articles. Each facet describes one aspect or dimension of the result articles and is associated with a category hierarchy for that aspect. The user can explore the result articles by specifying the desired conditions on multiple facets, *i.e.*, by simultaneously navigating through the category hierarchies.

Such a system must be automatic and dynamic, due to several reasons. First, given the sheer size and complexity of `Wikipedia`, a manual approach is prohibitively time-consuming. Second, it is difficult for a manual approach to scale and keep up to date with the fast growing contents on `Wikipedia`. Last but not least, query-dependent facets are necessary because keyword search results vary significantly across queries. In applications where faceted interfaces are deployed for relational tuples or metadata-rich objects (such as multimedia files or text corpus), the tuples/objects are captured by prescribed schemata with clearly defined dimensions (attributes), therefore a static faceted interface (either manually or automatically generated) would be effective. However, the articles in `Wikipedia` clearly do not have such pre-determined dimensions that can describe all possible dynamic query results. Thus efforts for static facets would be futile.

Automatic and dynamic facets discovery for `Wikipedia` is a significant and challenging undertaking. Even the notion of "facet" itself in `Wikipedia` does arise automatically. The concept of faceted interface is built upon two pillars: facets (i.e., dimensions or attributes) and the category hierarchy associated with each facet. Thus we need to answer questions such as: What are the dimensions or attributes of a `Wikipedia` article? Where does the category hierarchy on such an attribute come from?

Challenge 1: Defining facets on `Wikipedia` is non-trivial because the dimensions and hierarchies are not readily available.

The essence of our approach is to build upon the intensive structures (hyperlinks) and folksonomy (category system) created by collaborative Wikipedia authors. More specifically, we view hyperlinks within articles as pointers from articles to their attributes, and we exploit the category system in Wikipedia as the backbone of the facet hierarchies.

With regard to the concept of dimensions or attributes, the other Wikipedia articles hyperlinked from a search result article are treated as its attributes. The underlying intuition is simple. The fact that the authors of an article made hyperlinks to other articles is an indication of the significance of the linked articles in describing the former one. This view largely enriches the information associated with the result articles. For instance, in Example 1, Hungary is an attribute of Paul Erdos based on the hyperlink on the Paul Erdos article page. Therefore during exploration, the categories of Hungary are helpful in reaching Paul Erdos even though originally those categories may not be associated with Paul Erdos. The user could then navigate to Paul Erdos by specifying (via the country facet) that she is looking for an Hungarian Mathematician.

With regard to the concept of category hierarchy, the category system in Wikipedia provides the hierarchy of category-subcategory relationships for the categories on a dimension. For example, Liberal Democracies is a subcategory of Countries by form of an Government. Therefore the user can navigate by first specifying that she is looking for graph theoreticians based on their countries by the form of government and then further making the condition more specific by choosing Liberal Democracies.

Although the aforementioned concept is intuitive, discovering such facets automatically and dynamically presents a significant research challenge. Given a set of s result Wikipedia articles, there is a large number of attribute articles which in turn have many categories associated with complex hierarchical relationships. To just give a sense of the scale, in Wikipedia there are more than 2.5 million English articles with hundreds of millions of internal links, and close to half a million categories with several million category-subcategory relationships in the category system. From such a big search space, FacetedPedia needs to find a set of "good" facets, *i.e.,* a set of category hierarchies. Clearly the number of possible choices is overwhelmingly large, given the size of the search space. In addition, the problem is even more challenging because the utilities of multiple facets do not necessarily build up linearly: since each facet in the set should describe different aspects or dimensions of the result articles, a set of facets that are "good" individually may not be that "good" collectively if they overlap a lot with each other. Therefore we need to avoid facets with a large degree of overlapping.

Challenge 2: In order to find a set of k "good" facets from the huge space of possible facets, we must have effective metrics for measuring the "goodness" of facets both individually and collectively.

For addressing the above challenge, we propose a principled metric for ranking the facets individually, based on users' navigational cost in exploring the facets. Intuitively, a highly ranked facet should satisfy several criteria, for the ease of navigation: (1) *comprehensive*– it should reach as many result articles as possible; (2) *diversified*– it should reach the result articles through different categories in the hierarchy; (3) *shallow*– the hierarchy should not be very deep; and (4) *narrow*– the categories in the hierarchy should not have too many subcategories. Moreover, we design a similarity measure between facets based on the Jaccard coefficient [16] to capture the overlap between facets. Then we rank sets of k-facet collectively according to both average pairwise similarity and average navigational cost. In a purely ideal faceted interface, the individual facets should have smallest navigational costs and the multiple facets should not overlap with each other.

In discovering a faceted interface, we cannot possibly apply the above ranking functions exhaustively on all possible choices, due to the prohibitively large search space. Therefore we must design algorithms for discovering facets based on the ranking functions that avoid such exhaustive methods.

Challenge 3: Algorithms for discovering facets based on the ranking functions

In addressing this challenge, we focus on a subset of the search space where the facets do not contain unnecessary categories with respect to reaching the result articles. For ranking the facets individually, we design a recursive algorithm that is able to calculate the ranking scores of all the candidate facets in this space by one depth-first search (DFS) of the space of categories. For finding a "good" k-facet interface, we apply a hill climbing algorithm that looks for a local optimum, which is a k-facet interface that is better than all the neighboring k-facet interfaces (i.e., interfaces that can be obtained by replacing just one facet in the current interface), according to the ranking function for faceted interfaces.

1.3 Summary of Contributions and Outline of Paper

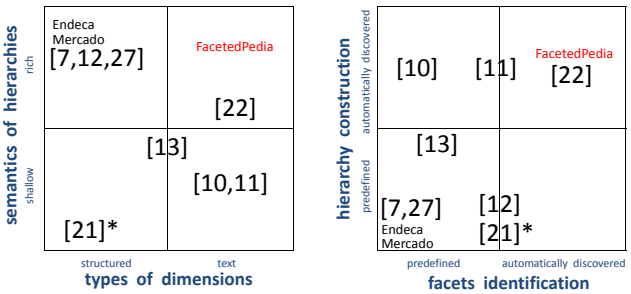
In summary, this paper makes the following contributions:

- **Concept: Faceted Wikipedia.** We propose an automatic and dynamic faceted retrieval system over Wikipedia. To the best of our knowledge, this is the first system of its kind for Wikipedia. The key concept is to view the Wikipedia articles hyperlinked from a given article as its attributes, and use the folksonomy (category system) as the backbone of facet hierarchies. (Section 3)
- **Metrics: Facets Ranking**. We propose ranking functions for measuring the "goodness" of facets, both individually (by navigational cost) and collectively (by combining costs and similarity). (Section 4)
- **Algorithms: Facets Discovery**. We design effective algorithms for discovering the facets based on the ranking functions. (Section 5)
- **System Evaluation: FacetedPedia**. We implemented and empirically evaluated the system. We conducted user study to compare with alternative approaches. (Section 6)

The rest of the paper is organized as follows. In Section 2 we compare FacetedPedia with other systems based on a taxonomy of faceted retrieval systems. In Section 3, we formally define the concept of faceted interface and the facets discovery problem in Facetedpedia. Section 4 discusses the metrics for ranking facets. We present the facets discovery algorithms in Section 5. Section 6 discusses some implemental details and the results of user study and experimental evaluation. We review the related work in Section 7. Section 8 concludes the paper.

2. EXISTING FACETED RETRIEVAL SYSTEMS: A COMPARATIVE STUDY

Several works studied the utility of faceted interface [17, 27, 14] and such an interface has been applied in various scenarios, including multimedia retrieval [27], text documents [14, 22, 11, 10], relational data [20, 21, 13], and metadata-rich text corpus [12,



(a) By the characteristics of the dimensions and the hierarchies.

(b) By the degree of automation and dynamism.

* The work does not support hierarchy on facets.

Figure 2: A taxonomy of faceted retrieval systems.

7]. Commercial vendors (*e.g.*, `Endeca`, `IBM`, and `Mercado`) also provide faceted navigation products for E-commerce applications. Commercial Websites such as `eBay.com` have adopted this facility. However, very few have investigated the problem of automatic and dynamic facets discovery and none has provided faceted interface over `Wikipedia`.

In this section we present two taxonomies to characterize various faceted retrieval systems and to compare them with `FacetedPedia` on two aspects, namely (a) the types of dimensions and the semantics of hierarchies in faceted interfaces; and (b) the degree of automation and dynamism in facets discovery.

(A) By Types of Dimensions and Semantics of Hierarchies:
In comparison with the facets in other systems, `FacetedPedia` sits between two extreme ends. In some systems the facets are structured attributes that come from schemata designed by domain experts and the attribute values are associated with domain-specific taxonomies. In such a scenario, facets are built upon predefined explicit structures and semantic information. In other systems a facet is a group of relevant textual terms from the retrieved documents, and the hierarchy over the terms is built upon thesaurus-based IS-A relationships (*e.g.*, [22]) or frequency-based subsumption relationships between general and specific terms (*e.g.*, [11, 10]). In such a scenario, the systems do not have enough structures and semantic information that can be leveraged. Our scenario and approach are unique: On the one hand, we build on top of the abundant hyperlink structures and semantics-rich category system that are created by the group intelligence of a large number of collaborative `Wikipedia` users; On the other hand, such existing information does not readily provide the facet dimensions and hierarchies for us, thus presenting a significant challenge in facets discovery.

The taxonomy based on this aspect is shown in Figure 2(a). With regard to the types of dimensions, some works provide facets on relational data (*e.g.*, `Endeca`, `Mercado`, [21]). The facets in [12, 7, 27] are on structured attributes although the objects may also have textual or semi-structured properties. Diederich et al. [13] created a faceted access interface for DBLP [3]. The dimensions are mostly structured properties (*e.g.*, *authors*, *year*), and the hierarchies on the dimensions are predefined based on domain knowledge. There is a special facet that provides a taxonomy summarizing the topic keywords in the searched bibliography. Finally, [22, 11, 10] support facets over text documents and each facet is a hierarchy on terms and phrases. Our work roughly belongs to the last category, since the hierarchies are on the titles of `Wikipedia` categories.

[3]http://dblp.l3s.de/

With regard to the semantic information captured in the hierarchies, `FacetedPedia` is based on `Wikipedia` folksonomy, and therefore incorporates rich semantic information into the facet hierarchies. In [11, 10] the hierarchies are built upon subsumption relationships between terms in documents, according to frequencies instead of semantics. In [10] they use external sources such as `Wikipedia` in extracting more candidate terms. In a document, a phrase matching the title of a `Wikipedia` article is a candidate term, and the titles of articles hyperlinked from that article are also candidate terms. This shares similar intuition with our model of hyperlinked articles as attributes. The `Castanet` algorithm [22] generates the hierarchy of terms by their IS-A relationships in the `WordNet`[4] lexical database. For those works where hierarchies are predefined ([27, 12, 7], `Endeca`, and `Mercado`), the hierarchies could be manually created, therefore could have rich semantic information. [13] is in the middle of Figure 2(a) since most of the dimensions are structured, except that the special topic taxonomy is based on the subsumption relationships between topic words.

(B) By Degree of Automation and Dynamism:
Discovering a full-fledged faceted interface involves two tasks: (1) *facets identification*– identifying what are the facets (*i.e.*, dimensions or attributes) and selecting the important facets; and (2) *hierarchy construction*– creating a hierarchy of categories on each facet. When accomplishing both tasks, manual versus automatic and static versus dynamic approaches could be applied.

The taxonomy based on this aspect is shown as Figure 2(b). None of the current facets retrieval systems could be effectively applied for `Facetedpedia`. First, there are no existing algorithms for dynamically discovering query-dependent facets. There are some works on selecting the most important facets at query time. However, their pool of candidate facets are predefined. Second, most algorithms are not fully automatic in both facets identification and hierarchy construction.

Commercial products from `Endeca` and `Mercado` do not automatically discover the facets or construct the hierarchy. They focus on providing the navigation support in predefined faceted interfaces. The publicly available research system `Flamenco` [27, 14] mostly focuses on user-interface issues instead of automatic facets generation. Debabrata et al. [12] studied the problem of selecting the most "interesting" facets for a repository of documents with structured properties, *i.e.*, the facets. The facets and their hierarchies are predefined, therefore there are no facets identification or hierarchy construction tasks. They do automatically select the most "interesting" facets, therefore we place this work in the middle along the dimension of facets identification. Ben-Yitzhak et al. [7] extended faceted search with aggregation and the techniques are for predefined facets. In [21] the authors studied facets search over relational tables. There is no need to identify the facets, which are relational attributes, although they do dynamically choose a subset of the facets for drilling down into the database. Moreover, there is no hierarchy associated with the attributes.

In [11] the set of facets are predefined, where each facet is associated with some seed terms. They automatically extract more candidate terms, insert them into the given facets, and create the hierarchies by subsumption. They also select the most important portion of the hierarchy for displaying in an limited space. In Diederich et al. [13] the special topic taxonomy is automatically generated.

With respect to the automaticity of facets discovery, the closest work to ours is the `Castanet` algorithm [22]. It automatically creates facets from the textual descriptions of a collection of items (*e.g.*, recipes). The hierarchies for the multiple facets are obtained

[4]http://wordnet.princeton.edu/

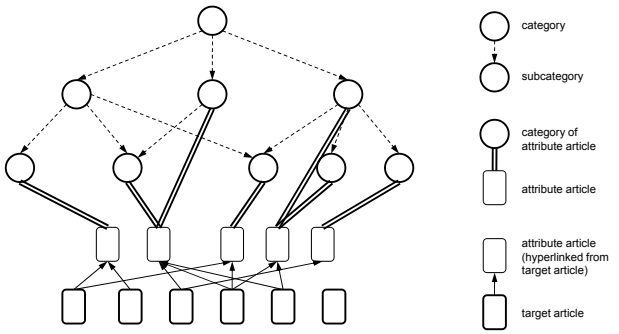


Figure 3: The concept of facet.

by first generating a single taxonomy of terms by IS-A relationships and then removing the root from the taxonomy. The algorithm is intended for short textual descriptions with limited vocabularies in a specific domain.

3. FACETED WIKIPEDIA

In this section, we first explain the intuition behind the concept of facets in our system and give formal definitions (Section 3.1). We then present the specification of facets discovery problem (Section 3.2).

3.1 Concepts of Facets over Wikipedia

The goal of FacetedPedia is to provide a faceted interface to navigate the result articles of a keyword search query to Wikipedia. The key in our definition of faceted interface is to view the articles hyperlinked from the result articles as their attributes, and to use the categories of the attribute articles as the facets.

Definition 1 (Target Article, Attribute Article): Given a keyword search query q to Wikipedia, the set of top-s ranked result articles, $\mathcal{T}=\{p_1, ..., p_s\}$, are the *target articles* of q.

Given a target article p, each Wikipedia article p' that is hyperlinked from p is an *attribute article* of p. This relationship is represented as $p \rightarrow p'$.

Given $\mathcal{T}$, the set of attribute articles is $\mathcal{A}=\{p'_1, ..., p'_m\}$, where each p'_i is an attribute article of at least one target article $p_j \in \mathcal{T}$. [5] ■

A Wikipedia article may belong to one or more *categories*. These categories are listed at the bottom of the article. There exists a *category hierarchy* that captures the supercategory and subcategory relationships between categories. The categories of articles and the category hierarchy are generated by users in the same collaborative fashion that articles are generated. The category hierarchy of Wikipedia articles can be viewed as a rooted and directed acyclic graph (DAG). [6] All the categories of articles are direct or indirect subcategories of the root, Category:Fundamental [1]. [7] [8]

[5] Note that target articles and attribute articles may overlap.

[6] Although cycles in the category hierarchy should usually be avoided as suggested by Wikipedia, there indeed exist cycles due to various reasons. Nevertheless, the graph can be made acyclic by detecting and removing a very small number of edges that represent low-quality or uncommon category-subcategory relationships. Section 6.1 discusses the details of cycle removal.

[7] An alternative root category is Category:Main Topic Classification. It is based on more detailed initial subcategories than Category:Fundamental is.

[8] There are categories for contents that are not articles. We do not consider these categories since we focus on articles. The overall root of the whole category hierarchy that contains all types of categories is Category:Contents.

Definition 2 (Category Hierarchy): The Wikipedia category hierarchy is a connected, rooted directed acyclic graph $\mathcal{H}(\mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$, where the node set $\mathcal{C}_\mathcal{H}=\{c\}$ is the set of categories and the edge set $\mathcal{E}_\mathcal{H}=\{c \dashrightarrow c'\}$ is the set of category-subcategory relationships between category c and subcategory c'. The root category of $\mathcal{H}$ is Category:Fundamental. ■

We define a *facet* as a rooted and connected subgraph of the category hierarchy.

Definition 3 (Facet): A *facet* $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$ is a rooted and connected subgraph of the category hierarchy $\mathcal{H}(\mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$, where $\mathcal{C}_\mathcal{F} \subseteq \mathcal{C}_\mathcal{H}$, $\mathcal{E}_\mathcal{F} \subseteq \mathcal{E}_\mathcal{H}$, and $r \in \mathcal{C}_\mathcal{F}$ is the root of $\mathcal{F}$. ■

The categories in the facet can "*reach*" the target articles $\mathcal{T}$ through attribute articles $\mathcal{A}$. That is, by following the category-subcategory hierarchy of the facet, we could find a category, then find an attribute article belonging to the category, and finally find some desirable target articles that have the attribute value. Using the typical concept of facet over structured attributes as an analogy, a target article p corresponds to an object or a tuple; an attribute article of p, p', corresponds to a value for a structured attribute of the object; finally the categories of p' and their supercategories correspond to the category hierarchy on the corresponding structured attribute. In order to capture the above notion of "reach", we formally define *category path* and *navigational path* as follows.

Definition 4 (Category Path): With respect to a facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, a *category path* in $\mathcal{F}$ is a sequence $c_1 \dashrightarrow ... \dashrightarrow c_t$, where,

- for $1 \le i \le t$, $c_i \in \mathcal{C}_\mathcal{F}$, *i.e.*, c_i is a category in $\mathcal{F}$;
- for $1 \le i \le t-1$, $c_i \dashrightarrow c_{i+1} \in \mathcal{E}_\mathcal{F}$, *i.e.*, c_{i+1} is a subcategory of c_i (in category hierarchy $\mathcal{H}$) and that category-subcategory relationship is kept in $\mathcal{F}$. ■

Definition 5 (Navigational Path): With respect to the target articles $\mathcal{T}$, the corresponding attribute articles $\mathcal{A}$, and a facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, a *navigational path* in $\mathcal{F}$ is a sequence $c_1 \dashrightarrow ... \dashrightarrow c_t \Rightarrow p' \leftarrow p$, where,

- $c_1 \dashrightarrow ... \dashrightarrow c_t$ is a category path in $\mathcal{F}$;
- $p' \in \mathcal{A}$, and c_t is a category of p' (represented as $c_t \Rightarrow p'$);
- $p \in \mathcal{T}$, and p' is an attribute article of p (*i.e.*, there is a hyperlink $p \rightarrow p'$).

The shortest possible navigational paths have only one category and have the form $c \Rightarrow p' \leftarrow p$.

Given a navigational path $c_1 \dashrightarrow ... \dashrightarrow c_t \Rightarrow p' \leftarrow p$, we say that the corresponding category path $c_1 \dashrightarrow ... \dashrightarrow c_t$ *reaches* target article p through attribute article p', and we also say that the category c_i (for any $1 \le i \le t$) *reaches* p through p'. Interchangeably we say that p is *reachable* from c_i (for any $1 \le i \le t$). ■

Note that in the above definition $p' \leftarrow p$ indicates that the direction of the hyperlink is from p to p'. However, the navigational paths do not follow hyperlinks. In fact, in our navigational model, a facet reaches the target articles through the attribute articles. Therefore we also say that the attribute article p' (directly) *reaches* the target article p. (Thus it is in the opposite direction of the hyperlink.) Interchangeably we say that p is (directly) *reachable* from p'.

For reaching the set of target articles with respect to a query q, a facet does not need to contain any category that cannot reach any target articles. In fact, having such categories in the facet is not only unnecessary but also harmful. A category that cannot reach any target articles is like a "deadend" in the facet (*i.e.*, no outgoing navigational paths to any attribute thus target articles). Exploring such a faceted interface, a user could be brought to the "deadend", resulting in a frustrating user experience. Therefore formally we have the following concept of *safe reaching facet*.

Definition 6 (Safe Reaching Facet): A facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$ *safely reaches* $\mathcal{T}$, the set of target articles, if $\forall c \in \mathcal{C}_\mathcal{F}$, there exists a target article $p \in \mathcal{T}$ such that c reaches p, *i.e.*, there exists $c \dashrightarrow ... \Rightarrow p' \leftarrow p$, a navigational path (and thus also a category path) of $\mathcal{F}$, starting from c, that reaches p. Such a $\mathcal{F}$ is a safe reaching facet of $\mathcal{T}$. ■

Note that it is not required for a (safe reaching) facet to fully reach every target article. In an environment such as relational databases, where there are prescribed rigorous schemata, it is feasible to require every tuple to have a non-null value on every attribute. However, in `Wikipedia` there are no such prescribed schemata, thus it is accepted that a facet cannot reach some of the target articles. However, it is indeed desired that a facet reaches a large percentage of the target articles. This criterion is incorporated into the facets ranking functions in Section 4.

Based on the above definition, we have the following straightforward Lemma 1.

Lemma 1: Given $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, a safe reaching facet of $\mathcal{T}$, every category path in $\mathcal{F}$ can reach at least one target article in $\mathcal{T}$, and thus every category path starting from r, the root of $\mathcal{F}$, can reach at least one target article in $\mathcal{T}$. (Because r has at least one category path to every category in $\mathcal{F}$, since it is rooted and connected.) ■

A faceted interface can then be defined as a set of safe reaching facets.

Definition 7 (Faceted Interface): Given a keyword query q, a faceted interface $I = \{\mathcal{F}_i\}$ is a set of safe reaching facets of the target articles $\mathcal{T}$. That is, $\forall \mathcal{F}_i \in I$, $\mathcal{F}_i$ safely reaches $\mathcal{T}$. ■

3.2 Problem Specification

Based on the formal definitions of faceted interface in Section 3.1, below is the specification of the problem that we study in this paper.

Specification of Facets Discovery Problem: Given the category hierarchy $\mathcal{H}(\mathcal{C}, \mathcal{E})$, for a keyword query q and its resulting target articles $\mathcal{T}$ and corresponding attribute articles $\mathcal{A}$, the *facets discovery problem* is to find the "best" faceted interface with k facets.

Given the sheer size and complexity of `Wikipedia`, the space of all possible k-facet interfaces is prohibitively large. Any rooted and connected subgraph of $\mathcal{H}$ that safely reaches $\mathcal{T}$ is a candidate facet, and any combination of k candidate facets would be a candidate faceted interface. Given the huge search space, we would need ranking metrics in comparing candidate faceted interfaces, and we also need a search algorithm that searches the space for the best interface, optimizing for the metrics. We shall introduce the ranking metrics in Section 4 and the search algorithms in Section 5.

4. FACETS RANKING METRICS

As motivated in Section 1.2 and Section 3.2, we must define effective ranking metrics for measuring the "goodness" of facets both individually and collectively, in order to find the "best" k-facet interface. We develop the single-facet and multi-facet ranking functions, in Section 4.1 and 4.2, respectively.

4.1 Single-Facet Ranking

Intuitively, a highly ranked facet should satisfy the following criteria:

- *Comprehensive*: The facet should reach as many target articles as possible.
- *Diversified*: The facet should cover target articles through different attribute articles.
- *Narrow*: The number of subcategories for a category should be small.
- *Shallow*: The number of steps in reaching target articles should be small.

Note that these criteria in fact may compete with each other. For example, a more comprehensive facet may tend to be wider and deeper.

Based on the above intuition, we propose a single-facet ranking function based on user's navigational cost in using a facet.[9] We model users' navigational behaviors as follows and define the cost function accordingly.

User Navigation Model: Here we assume a single facet is provided to a user for navigation (multi-facets navigation is discussed later in Section 4.2. For any facet, the user starts her navigation from the root. At each step when the user has reached a category node, she examines the set of subcategories available and either chooses one of the subcategories, or chooses one of the attribute pages. The navigation terminates when the selected attribute page is an attribute of a target article page.

Our cost model is designed to capture the effort undertaken by the user in both browsing the subcategories at each category node, as well as following links to subcategories and attribute pages. In the actual `Facetedpedia` system, the user is also allowed to retract her search by going backwards up the hierarchy, but in our cost formula we only consider the forward cost.

Intuitively, we compute the cost of a single facet as the average cost of any path taken from the root to a reachable target page. In measuring the cost of a path, it should be intuitively clear that a low-cost path should be short in length (shallowness criterion), and have small number of sub-categories at each category node to reduce browsing cost (narrowness criterion).

However, the navigational cost would not be able to capture the aforementioned criterion on comprehensiveness, that is, the ability of a facet to reach as many target articles as possible. As discussed in Section 3.1 a facet may not be able to reach all the target articles. Missing the unreachable articles presents an unsatisfactory user experience. Thus, when defining the overall cost model of a facet, we set a (parameterized) high penalty for such unreachable articles, and combine this cost with the average cost of all paths that are able to reach target articles.

Now we formally define the single-facet ranking function.

Definition 8 (Reachable Target Articles): A target article $p \in \mathcal{T}$ is *reachable* from a facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$ if r, the root of $\mathcal{F}$, can reach p, *i.e.*, there exists a navigational path $r \dashrightarrow ... \Rightarrow p' \leftarrow p$, where p' is an attribute article of p.

With respect to the target articles $\mathcal{T}$ and the facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, the reachable target articles from $\mathcal{F}$ is $\mathcal{T}_r = \{p | p \in \mathcal{T} \wedge p \text{ is reachable from } \mathcal{F}\}$. Without confusion, we use $\mathcal{F}$, $\mathcal{F}_r$ and $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$ interchangeably, and use $\mathcal{T}_r$ and $\mathcal{T}_{\mathcal{F}_r}$ interchangeably. ■

Definition 9 (Cost of Navigational Path): With respect to the target articles $\mathcal{T}$, the corresponding attribute articles $\mathcal{A}$, and a facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, the cost of a navigational path in $\mathcal{F}$, $l = c_1 \dashrightarrow ... \dashrightarrow c_t \Rightarrow p' \leftarrow p$, is defined as

$$cost(l) = fanout(p') + \sum_{c \in \{c_1, ..., c_t\}} fanout(c) \quad (1)$$

In Equation 1, $fanout(p')$ is the fanout of attribute article p',

$$fanout(p') = |\mathcal{T}_{p'}| \quad (2)$$

[9]Note that [21] also selects facets based on navigational costs, although their system is of different nature, as discussed in Section 2.

where $\mathcal{T}_{p'}$ is the set of (directly) reachable target articles from p',

$$\mathcal{T}_{p'} = \{p | p \in \mathcal{T} \wedge p \rightarrow p' (i.e., there\ is\ a\ hyperlink\ from\ p\ to\ p'\} \quad (3)$$

In Equation 1, $fanout(c)$ is the fanout of category c in facet $\mathcal{F}$,

$$fanout(c) = |\mathcal{A}_c| + |\mathcal{C}_c| \quad (4)$$

where $\mathcal{A}_c$ is the set of attribute articles that belong to category c,

$$\mathcal{A}_c = \{p' | p' \in \mathcal{A} \wedge c \Rightarrow p'\} \quad (5)$$

and $\mathcal{C}_c$ is the set of subcategories of c in $\mathcal{F}$,

$$\mathcal{C}_c = \{c' | c' \in \mathcal{C}_\mathcal{F} \wedge c \dashrightarrow c' \in \mathcal{E}_\mathcal{F}\} \quad (6)$$ ■

Definition 10 (Cost of Facet): With respect to the target articles $\mathcal{T}$, the cost of a safe reaching facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, $cost(\mathcal{F}_r)$, is a weighted sum of the coverage ratio of $\mathcal{F}$ and the average cost of the navigational paths starting from r (in logarithmic form), defined as

$$cost(\mathcal{F}_r) \leftarrow log(avg\_path\_cost) + penalty \times \frac{|\mathcal{T}| - |\mathcal{T}_r|}{|\mathcal{T}_r|} \quad (7)$$

In Equation 7, avg_path_cost is

$$avg\_path\_cost = \frac{\sum_{l \in NavPath} cost(l)}{|NavPath|} \quad (8)$$

where $NavPath$ is the set of navigational paths in $\mathcal{F}$ from r,

$$NavPath = \{l | l = r \dashrightarrow ... \Rightarrow p' \leftarrow p\} \quad (9)$$

In Equation 7, $\frac{|\mathcal{T}|-|\mathcal{T}_r|}{|\mathcal{T}_r|}$ is the ratio between the number of unreachable target articles ($|\mathcal{T}| - |\mathcal{T}_r|$) and the number of reachable target articles ($|\mathcal{T}_r|$). The parameter $penalty$ is for the purpose of penalizing a facet of not being able to reach some target articles. ■

The value of penalty is empirically selected (Figure 4) by investigating the relationship between the number of uncovered target articles and the average path cost of a candidate facet node. Given the linear function on $log(avg\_path\_cost)$ and $\frac{|\mathcal{T}|-|\mathcal{T}_r|}{|\mathcal{T}_r|}$, we find the slope (*i.e.*, $penalty$) that is able to separate the "best/good" facets from "poor/worst" facets. Empirically we set that $penalty$ value to be 7, which is the value used in our system evaluation.

The cost function based on navigational cost matches the aforementioned four criteria.

- *Comprehensive*: We penalize a facet for unreachable target articles.
- *Diversity*: A facet is not diversified if a single category or attribute article can reach many target articles. Such a facet would have higher cost due to the larger fanouts associated with the category and attribute articles.
- *Narrowness*: The facet with small fanouts on the nodes would be narrow.
- *Shallowness*: The facet with small number of steps in reaching target articles would involve less number of steps in navigation.

4.2 Multi-Facet Ranking

We now assume the user is provided with k-facets for simultaneous navigation. For any facet, the user starts her navigation from the root and goes down the category hierarchy or chooses one of the attribute pages. The navigation terminates when enough attribute pages have been reached that together isolate target article pages.

Even though the cost function in Section 4.1 can rank individual facets, measuring the "goodness" of a faceted interface, *i.e.*, a set

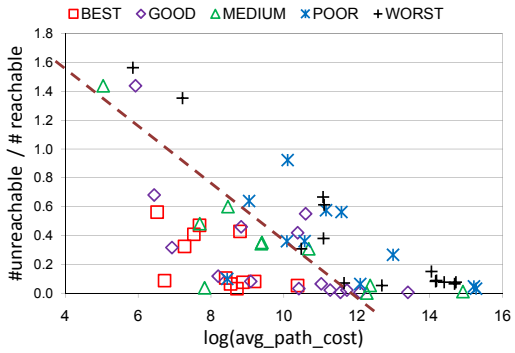


Figure 4: Empirical selection of value for parameter $penalty$.

of facets, is not straightforward. This is because the best k-facet interface may not be simply the top-k facets ranked according to the single-facet cost function (in other words, the facets that are best individually may not be the best together). The reason is clear, as the following example shows. In some circumstances, the top-2 facets may have a parent-child relationship. However, presenting them together as a faceted interface would not be desirable since they overlap significantly, thus cannot capture the properties of target articles through multiple dimensions.

In principle one can extend the navigation cost model to also apply for navigating multiple facets simultaneously. However, such modeling of the navigational cost of multiple facets is very complex. Therefore we simplify the metric by introducing the notion of *average similarity* between the k-facets.

Definition 11 (Best K-Facets): The best k-facets is the set of k facets with the smallest average pair-wise similarity. The formula is shown below,

$$score(\mathcal{I} = \{\mathcal{F}_1, ..., \mathcal{F}_k\}) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} sim(\mathcal{F}_i, \mathcal{F}_j), \quad (10)$$

where $sim(\mathcal{F}_i, \mathcal{F}_j)$ is defined by Jaccard coefficient, *i.e.*,

$$sim(\mathcal{F}_i, \mathcal{F}_j) = \frac{\mathcal{C}_{\mathcal{F}_i} \bigcap \mathcal{C}_{\mathcal{F}_j} + \mathcal{A}_{\mathcal{F}_i} \bigcap \mathcal{A}_{\mathcal{F}_j}}{\mathcal{C}_{\mathcal{F}_i} \bigcup \mathcal{C}_{\mathcal{F}_j} + \mathcal{A}_{\mathcal{F}_i} \bigcup \mathcal{A}_{\mathcal{F}_j}} \quad (11)$$

$$\mathcal{A}_{\mathcal{F}_i} = \{p' | p' \in \mathcal{A} \wedge \exists c \in \mathcal{C}_{\mathcal{F}_i}\ s.t.\ c \Rightarrow p'\} \quad (12)$$ ■

where $\mathcal{A}_{\mathcal{F}_i}$ is the set of attribute articles reachable from $\mathcal{F}_i$ as defined above.

Finding the best k-facets appears to be a difficult optimization problem. The naive technique would require us to examine all combinations of k-facets to determine the one with the lowest cost. However, this approach would be exceedingly slow, and therefore in section 5.3, we discuss several principled heuristics to solve this problem including greedy heuristics as well as hill climbing techniques.

5. AUTOMATIC AND DYNAMIC FACETS DISCOVERY ALGORITHMS

With the single-facet and multi-facet ranking functions developed in Section 4, one straightforward approach in discovering k-facet interface is to enumerate all possible k-facet interfaces with respect to the category hierarchy $\mathcal{H}$ and apply the facets ranking functions directly. Such a naïve method results in the exhaustive

Algorithm 1: Facets Discovery

Input:
q: a keyword search query;
$\mathcal{H}(\mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$: category hierarchy;
$penalty$: parameter for penalizing facets with poor coverage of target articles.

Output:
I_k: a discovered faceted interface with k facets

1 $\mathcal{T} \leftarrow$ the top-s ranked results (`Wikipedia` articles) for query q
2 Algorithm 2 `// Get Attribute Articles (`$\mathcal{A}$`) and Relevant Category Hierarchy (`$\mathcal{RCH}(\mathcal{C}_{\mathcal{RCH}}, \mathcal{E}_{\mathcal{RCH}})$`).`
3 Algorithm 3 `// Rank Individual Facets.`
4 Algorithm 4 `// Select a` k`-Facet Interface.`
5 **return** $\mathcal{I}_k$

examination of all possible combinations of k facets from all possible facets, *i.e.*, rooted and connected subgraphs of $\mathcal{H}$. Clearly that is a prohibitively large search space, therefore the na'ıve technique would be extremely costly.

Note that in this k-facet discovery problem, there are two search spaces. One is the space of facets and the other is the space of k-facet interfaces, which are combinations of facets in the first space. Therefore in order to address the challenges, our solutions hinge on (a) shrink the two search spaces (Section 5.1); and (b) develop effective algorithms in searching the spaces (Section 5.2 and 5.3).

5.1 Relevant Category Hierarchy ($\mathcal{RCH}$) (Algorithm 2)

To shrink the search space, we first focus on a subset of the exhaustive search space that guarantees to contain all the safe reaching facets. As discussed in Section 3.1, a facet that is not a safe reaching facet is harmful as it brings users to the "deadend". Furthermore, such a facet is completely unnecessary because, by Definition 7, clearly there is always a corresponding safe reaching facet that sustains all the navigational paths in the unsafe one and yet do not have any unnecessary categories. Therefore this shrinking of space is lossless. The shrunk space is called *relevant category hierarchy* ($\mathcal{RCH}$), defined as follows.

Definition 12 (Relevant Category Hierarchy): With respect to the category hierarchy $\mathcal{H}(\mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$, the set of target articles $\mathcal{T}$, and the corresponding attribute articles $\mathcal{A}$, the *relevant category hierarchy* ($\mathcal{RCH}$) of $\mathcal{T}$ is a subgraph of $\mathcal{H}$. Given any category in $\mathcal{RCH}$, it is either directly a category of some attribute article $p' \in \mathcal{A}$ or a supercategory or ancestor of such categories that p' belongs to. The edges in $\mathcal{RCH}$ include all the category-subcategory relationships among the categories in $\mathcal{RCH}$. ■

The procedural algorithm for getting $\mathcal{RCH}$ is shown in Algorithm 2. Based on the definition, it is clear that we have the following Corollary 1, which states that $\mathcal{RCH}$ is lossless in the sense that it contains all the safe reaching facets.

Corollary 1: Every safe reaching facet of the target articles $\mathcal{T}$ is a (rooted and connected) subgraph of $\mathcal{RCH}$. ■

However, the reverse of Corollary 1 is not true. That is, not every rooted and connected subgraph of $\mathcal{RCH}$ is a safe reaching facet. Therefore, even though $\mathcal{RCH}$ is much smaller than the original search space, $\mathcal{H}$, it can be still very large. In building `FacetePedia`, we find that for 200 target articles, there could be

Algorithm 2: Construct Relevant Category Hierarchy and Get Attribute Articles

Input:
$\mathcal{T}$: target articles;
$\mathcal{H}(\mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$: category hierarchy;

Output:
$\mathcal{A}$: attribute articles;
$\mathcal{RCH}(\mathcal{C}_{\mathcal{RCH}}, \mathcal{E}_{\mathcal{RCH}})$: relevant category hierarchy.

`// get attribute articles.`
1 $\mathcal{A} \leftarrow \phi$; $\mathcal{C}_{\mathcal{RCH}} \leftarrow \phi$; $\mathcal{E}_{\mathcal{RCH}} \leftarrow \phi$
2 **foreach** $p \in \mathcal{T}$ **do**
3 **foreach** $p \rightarrow p'$, *i.e., a hyperlink from* p *to* p' **do**
4 $\mathcal{A} \leftarrow \mathcal{A} \cup \{p'\}$
`// start from the categories of attribute articles.`
5 **foreach** $p' \in \mathcal{A}$ **do**
6 **foreach** $c \Rightarrow p'$, *i.e., a category that* p' *belongs to* **do**
7 $\mathcal{C}_{\mathcal{RCH}} \leftarrow \mathcal{C}_{\mathcal{RCH}} \cup \{c\}$
`// get` RCH `by recursively obtaining the supercategories of obtained categories.`
8 $\mathcal{C} \leftarrow \mathcal{C}_{\mathcal{RCH}}$
9 **while** $\mathcal{C}$ *is not empty* **do**
10 $\mathcal{C}' \leftarrow \phi$
11 **foreach** $c \in \mathcal{C}$ **do**
12 **foreach** $c' \dashrightarrow c \in \mathcal{E}_\mathcal{H}$ **do**
13 **if** $c' \notin \mathcal{C}_{\mathcal{RCH}}$ **then**
14 $\mathcal{C}_{\mathcal{RCH}} \leftarrow \mathcal{C}_{\mathcal{RCH}} \cup \{c'\}$
15 $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{c'\}$
16 $\mathcal{E}_{\mathcal{RCH}} \leftarrow \mathcal{E}_{\mathcal{RCH}} \cup \{c' \dashrightarrow c\}$
17 $\mathcal{C} \leftarrow \mathcal{C}'$
18 **return** $\mathcal{A}$ *and* $\mathcal{RCH}(\mathcal{C}_{\mathcal{RCH}}, \mathcal{E}_{\mathcal{RCH}})$

thousands of corresponding attribute articles and the $\mathcal{RCH}$ can contain thousands of categories, which much large number of category-subcategory relationships.

Due to the above reason, we further shrink the space by considering one type of safe reaching facets, called $\mathcal{RCH}$*-induced facets*.

Definition 13 ($\mathcal{RCH}$-Induced Facet): Given the relevant category hierarchy $\mathcal{RCH}$ of the set of target articles $\mathcal{T}$, a facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$ is $\mathcal{RCH}$*-induced* if it is a rooted induced subgraph of $\mathcal{RCH}$, *i.e.*, all the descendants of the root r and their category-subcategory relationships are retained from $\mathcal{RCH}$. ■

The following theorem can be proven. We omit the simple proof due to space limitations. Theorem 1 guarantees that when searching in the new smaller search space ($\mathcal{RCH}$-induced facets), we would only find those safe reaching facets in $\mathcal{RCH}$.

Theorem 1: Every $\mathcal{RCH}$*-induced facet* is a safe reaching facet. ■

Note that it is not true that every safe reaching facet is $\mathcal{RCH}$-induced. Therefore we can no longer guarantee that every safe reaching facet is included in this new space. By that sense this space is lossy. Intuitively the $\mathcal{RCH}$-induced facets are those "maximal" safe reaching facets, *i.e.*, they contain all the subcategories and category-subcategory relationships under the roots. It is very possible that a "good" facet is not such a "maximal" facet. This forms a very challenging and worthwhile research problem for future study.

The outline of our k-facet discovery algorithm is in Algorithm 1. It first obtains $\mathcal{RCH}$ (Algorithm 2), then applies a depth-first search (DFS) algorithm to rank the facets individually (Algorithm 3), and

Algorithm 3: Facets Ranking

Input:
$\mathcal{T}$: target articles;
$\mathcal{AC}$: list of (article, category) pairs;
$\mathcal{A}$: attribute articles;
$\mathcal{RCH}(\mathcal{C}_{\mathcal{RCH}}, \mathcal{E}_{\mathcal{RCH}})$: relevant category hierarchy;
$penalty$: parameter for penalizing facets with poor coverage of target articles.

Output:
$\mathcal{I}_n$: the top n (complete) facets with the smallest costs.

// For each attribute article, get its reachable target articles.
foreach $p' \in \mathcal{A}$ **do**
  $\mathcal{T}_{p'} \leftarrow \{p | p \in \mathcal{T} \wedge \exists\ p \rightarrow p'$ (hyperlink from p to $p')\ \}$
  $fanout(p') \leftarrow |\mathcal{T}_{p'}|$
// For each category (thus the corresponding complete facet), get its cost.
foreach $c \in \mathcal{C}_{\mathcal{RCH}}$ *in reverse topological order* **do**
  // directly reachable attribute articles.
  $\mathcal{A}_c \leftarrow \{p' | p' \in \mathcal{A} \wedge (p', c) \in \mathcal{AC}\}$
  // subcategories.
  $\mathcal{C}_c \leftarrow \{c' | (c, c') \in \mathcal{E}_{\mathcal{RCH}}\}$
  // reachable target articles.
  $\mathcal{T}_c \leftarrow (\cup_{p' \in \mathcal{A}_c} \mathcal{T}_{p'}) \bigcup (\cup_{c' \in \mathcal{C}_c} \mathcal{T}_{c'})$
  $fanout(c) \leftarrow |\mathcal{A}_c| + |\mathcal{C}_c|$
  $\#path(\mathcal{F}_c) \leftarrow |\mathcal{A}_c| + \sum_{c' \in \mathcal{C}_c} \#path(\mathcal{F}'_c)$
  $cost_1 \leftarrow \sum_{c' \in \mathcal{C}_c} [(cost(\mathcal{F}'_c) + fanout(c)) \times \#path(\mathcal{F}'_c)]$
  $cost_2 \leftarrow \sum_{p' \in \mathcal{A}_c} [(fanout(p') + fanout(c)) \times fanout(p'_i)]$
  $avg\_path\_cost \leftarrow \frac{cost_1 + cost_2}{\#path(\mathcal{F}_c)}$
  $cost(\mathcal{F}_c) \leftarrow avg\_path\_cost + penalty \times \frac{|\mathcal{T}| - |\mathcal{T}_c|}{|\mathcal{T}_c|}$
sort $\mathcal{C}_{\mathcal{RCH}}$ (*i.e.*, the complete facets) by their costs
$\mathcal{I}_n \leftarrow$ the top n (complete) facets with the smallest costs.
return $\mathcal{I}_n$

Algorithm 4: Facets Selection

Input:
$\mathcal{I}_n$: the top n (complete) facets with the smallest costs.

Output:
I_k: a discovered faceted interface with k facets ($k<n$)

// hill climbing: start from a k-facet interface, at each step go to a neighbor k-facet interface with both lower average cost and smaller average pairwise similarity.
$\mathcal{I}_k \leftarrow$ a random k-facet subset of $\mathcal{I}_n$
$\mathcal{I}' \leftarrow \mathcal{I}_n - \mathcal{I}_k$
repeat
  make $\mathcal{I}_k = <\mathcal{I}_k[1], ..., \mathcal{I}_k[k]>$ sorted in increasing order of cost
  make $\mathcal{I}' = <\mathcal{I}'[1], ..., \mathcal{I}'[n-k]>$ sorted in increasing order of cost
  for $i = k$ **to** 1 **step** -1 **do**
    for $j = 1$ **to** $n-k$ **do**
      $\mathcal{I}_{new} \leftarrow \mathcal{I}_k \cup \{\mathcal{I}'[j]\} - \{\mathcal{I}_k[i]\}$
      $avgS_1 \leftarrow \frac{\sum_{\mathcal{F}_c, \mathcal{F}_{c'} \in \mathcal{I}_k, \mathcal{F}_c \neq \mathcal{F}_{c'}} sim(\mathcal{F}_c, \mathcal{F}_{c'})}{k(k-1)/2}$
      $avgC_1 \leftarrow \sum_{\mathcal{F}_c \in \mathcal{I}_k} cost(\mathcal{F}_c)$
      $avgS_2 \leftarrow \frac{\sum_{\mathcal{F}_c, \mathcal{F}_{c'} \in \mathcal{I}', \mathcal{F}_c \neq \mathcal{F}_{c'}} sim(\mathcal{F}_c, \mathcal{F}_{c'})}{k(k-1)/2}$
      $avgC_2 \leftarrow \sum_{\mathcal{F}_c \in \mathcal{I}'} cost(\mathcal{F}_c)$
      if $avgS_1 < avgS_2$ **and** $avgC_1 < avgC_2$ **then**
        $\mathcal{I}_k \leftarrow \mathcal{I}'$
        $\mathcal{I}' \leftarrow \mathcal{I}_n - \mathcal{I}_k$
        go to line 5
until $\mathcal{I}_k$ *does not change* ;
return $\mathcal{I}_k$

finally applies greedy algorithm and hill-climbing algorithm to rank k-facet interfaces (Algorithm 4).

5.2 Ranking the Facets (Algorithm 3)

For ranking the facets individually, we design a recursive algorithm that is able to calculate the ranking scores of all the candidate facets by one depth-first search (DFS) of the space of categories. The detailed algorithm is shown in Algorithm 3. The essence is that by depth first search, we could calculate the cost of the facets (rooted at the categories that are traversed) in postorder traversal. The cost of a facet associated with its root category is accumulated from the costs of the associated attributes articles and the costs of the facets associated with its subcategories.

5.3 Selecting the k-Facet Interface (Algorithm 4)

During our research, we investigate several techniques to choose k-facet selection algorithm. Below, we briefly describe the 4 different techniques that we have investigated and evaluated by experimentation. Each of these k-Facet selection algorithm works on the selected top-200 nodes of our Single-Facet ranking algorithm discussed above.

- Top-k Facet : In this method we just consider the Top-k facets that has been generated by our Single Facet selection algorithm.
- Greedy Incremental Method : In this method, we start with the node which has highest rank. At each step, the next node is chosen by greedily by looking into the neighbors of the already selected node in such a way that this node is least similar with its neighbor.
- Hill-climbing with one random start considering only similarity: In this method we try to find out one local optima of k-set of facets depending on the lowest similarity value. The process starts by selecting a random set of k nodes. At each step, the node with lowest rank is tried to be replaced by its immediate neighbor (one with the highest rank) using simple hill climbing if that replacement decreases the similarity value more. The process is stopped when similarity reaches a local optima.
- Hill-climbing with one random start combining rank and similarity : In this method we try to find out one local optima of k-set of facets depending on the combination of similarity and ranking values . The process starts by selecting a random set of k nodes. At each step, the node with lowest rank is tried to be replaced by its immediate neighbor (one with the highest rank) using simple hill climbing if that replacement decreases the combined value more. The process is stopped when it reaches a local optima.

6. EXPERIMENTAL EVALUATION

In this section we describe some of the implementation details of our preprocessing task, our experimental setup and the quality of

the output generated by Single Facet Ranking and Multiple Facet Ranking algorithms in FacetedPedia. Quality is evaluated by setting up user study in Amazon Mechanical Turk [10] . In addition to that, quality is also measured by different characteristics of a facet, such as depth, coverage etc (as discussed in Section 4).

6.1 Preprocessing of Wikipedia raw data

We download different raw Wikipedia dataset from Wikimedia. [11]. Different preprocessing tasks are performed on the dataset including cleaning, making its formats consistent etc. In addition to that, Wikipedia category hierachy dataset has a small number of cycles which was introduced by the collaborative modification of wikipedia articles by different editors. As one of our preprocessing tasks, we detect all such simple cycles in the wikipedia category hierarchy (total number of detected simple cycle is 594). Cycles are detected by using a modified Depth First Search algorithm. In order to preserve the richness of the category hierarchy, we manually delete one directed edge from each such cycles by selecting the one which semantically deviates from correct directionality

6.2 Benchmark Queries

We have taken 21 keyword queries for our evaluation purpose. The queries are as follows: 1)Nobel Laureates 2)Action Movies 3)Country Singers 4)European Physicists 5)American Presidents 6)Nobel Prize Winners 7)Mountain Ranges of Asia 8)Turing Award Winners 9)Ivy League Schools 10)Philosopher 11)American Civil War 12)Internet 13)Economic Depression 14)database research 15)national parks in United States 16)retirement plan in United States 17)Chinese cuisine 18)passenger cars 19)villains Batman 20)superheroes product 21)Global Warming. All evaluations are based upon subsets of these queries.

Next, we describe the evaluation of quality of the interface generated by FacetedPedia with one of the related work Castanet [22]. This evaluation is obtained by doing user study in Amazon Mechanical Turk.

6.3 User Study: Effectiveness of the Generated Faceted Interfaces

Five of our benchmark queries are evaluated by Mechanical Turk users for Single-Facet Interface and Multi-Facet Interface. Effectiveness is evaluated by a user by assigning a quantitative score, such as, 1) Not Relevant 2)Relevant, or 3) Very Relevant.

6.3.1 Single-Facet Study

In this set of studies, Each task contains 1 query. Five different such tasks are asked. For each query and a target article, one path from FacetedPedia vs. 1 path from Castanet are compared. A user is asked to evaluate each of these two paths individually. She is also required to evaluate the absolute preference of one over the other. At this context, we would like to note that Castanet [22] system is used for generating facets from a static collection of documents. Here, we apply that algorithm on the dynamic set of pages which are the results of a keyword query. Although not originally designed for such purposes, Castanet still appears to be the possible closest related work with FacetedPedia. As observed from Figures 7 and 5 , for Single-Facet study, FacetedPedia is unanimously preferred over Castanet and has been evaluated to be Very Relevant most of the time.

6.3.2 Multi-Facet Study

In this set of studies, each task contains 5 queries from our set of benchmark queries. For each query and a target article, a set of 3 paths (from three different facets) from FacetedPedia vs. another set of 3 different paths (from three different facets) from Castanet are compared. Similarly, a user is asked to evaluate each of these two set of paths individually. She is also required to evaluate the absolute preference of one set over the other. Similarly, as observed from Figures 9 and 10, for Multiple-Facet paths, FacetedPedia is unanimously preferred over Castanet and has been evaluated to be Very Relevant most of the time.

Next, we describe our internal evaluations for measuring the quality of the Facets obtained from FacetedPedia.

6.4 Quality of the Faceted Interface

In this subsection, we evaluate the quality of the generated facets by FacetedPedia by measuring their depth and coverage. We want to note here that according to our discussion in Section-4, higher coverage is better and shallow facets are good. In addition to that, we also show a snippet of our Top-1 facet, generated by our Single-Facet algorithm for the query "Villain Batman".

6.4.1 Depth, Coverage of facets

Figure 13 shows the average depth of a facet. As it can be seen from Figure, in most of the times our generated facets are shallow. Figure 12 shows the coverage of our generated facets. The coverage is measured by the number of the target articles that a particular facet can reach. In all of our experiments, we set the parameter of number of valid returned articles to be 200. Figure 12 validates that the returned facets by FacetedPedia have indeed good coverage.

6.4.2 Example Full Facet

Snippet of a Facet generated for query "Villains Batman" is shown in Figure 11.

6.5 Effectiveness of Ranking Function

In this set of experiments, we aim to evaluate the effectiveness of our ranking function for Single-Facet ranking. We perform a user study by comparing the root of Top-10 facets according to our ranking function with any 10 randomly picked up facet's roots. Each task in the Mechanical Turk here shows 10 facet roots, either ranked by our ranking function or chosen randomly. A user is asked to give a quantitative evaluation of each of these sets, by specifying either 1) Not Relevant, 2)Relevant or 3) Very Relevant. Four different queries are asked for each such tasks. The results from all four queries are aggregated and shown in Figure 14 and Figure 15.

As seen from Figures 14 and 15, the effectiveness of top-10 facets returned by FacetdPedia is evaluated much higher most of the time than that of the set of Random-10 facets. Next, we perform user study on the quality of our various K-Facet selection algorithms.

6.6 User Study: Various k-facet selection algorithms

In this subsection, we perform user study on various K-Facet Selection algorithms. In particular, to choose a set of top-K facets we apply 4 different algorithms (as discussed in Subsection 4) on the top-200 facets chosen by Single-Facet ranking function.

In order to evaluate the quality of 4 different K- Facet Selection algorithms defined in Subsection 4, we perform the following user study in Amazon Mechanical Turk - Each task here contains one query and a set of 3 paths from one of these 4 algorithms. For a given query with a target article, a user is asked to evaluate that set of paths by scoring it to be 1) Not Relevant, 2)Relevant or 3)Very relevant.

Figure 16 shows the aggregated user feedback on those 4 algorithms. As can be seen from the figure, among methods $(1) - (4)$, Top-10 comes out to be the least effective by user study. Among Hill Climbing with similarity and Hill Climbing with combination of similarity and rank, both receive good feedback. However as a comparison, the latter turns out to be more effective by user study. This concludes our discussion on experimental evaluation.

[10] www.mturk.com

[11] http://download.wikimedia.org

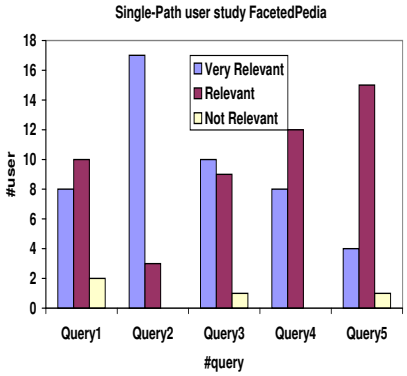


Figure 5: User Study of Single-Facet Path by FacetedPedia

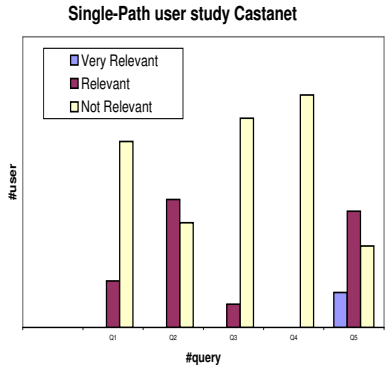


Figure 6: User Study of Single-Facet Path by Castanet

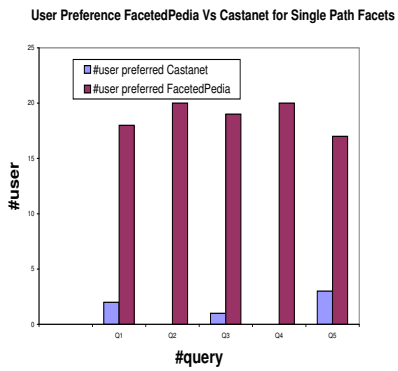


Figure 7: User preference of FacetedPedia over Castanet for Single-Facet Path

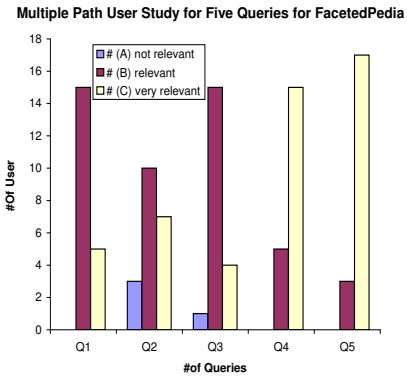


Figure 8: User Study of Multiple Facet Path by FacetedPedia

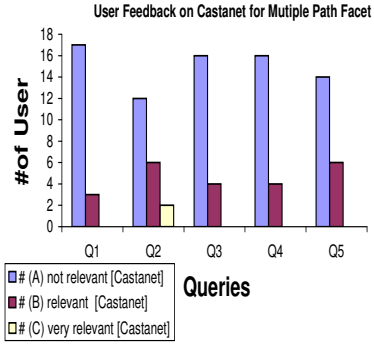


Figure 9: User Study of Multiple Facet Path by Castanet

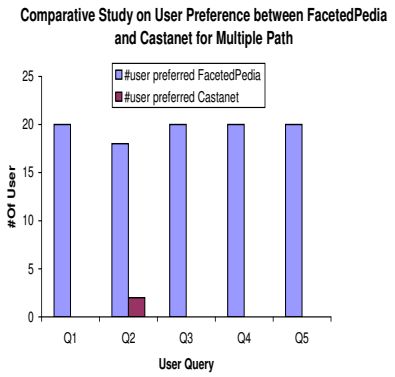


Figure 10: User preference of FacetedPedia over Castanet for Multiple Facet Path

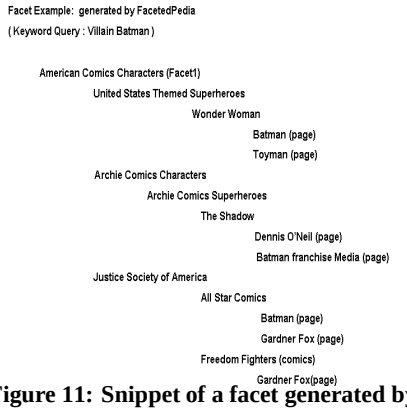


Figure 11: Snippet of a facet generated by FacetedPedia for query Villains Batman

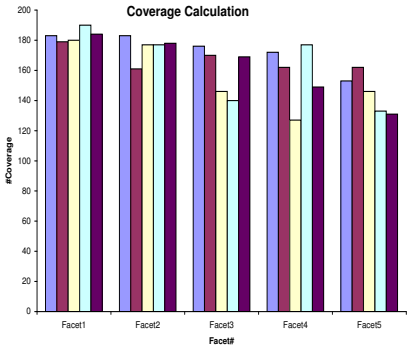


Figure 12: Coverage Calculation of 5 Facets for 5 queries

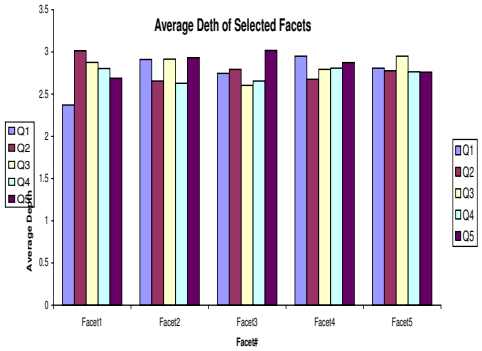


Figure 13: Average Depth of 5 Facets for 5 Queries

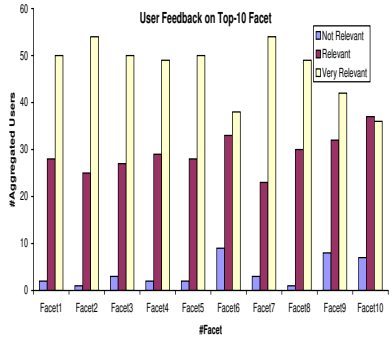


Figure 14: User Study on Top-10 returned facets by Single-Facet Ranking Function in FacetedPedia

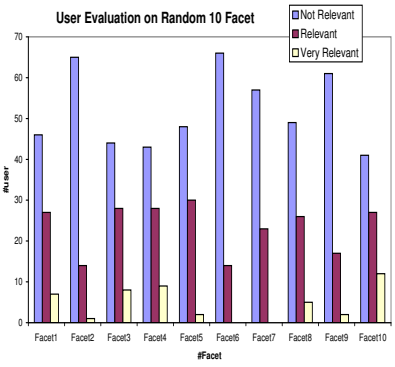


Figure 15: User Study on Random-10 Facets

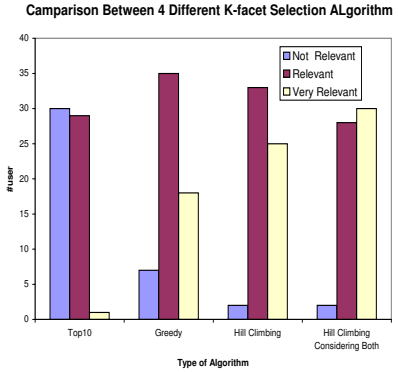


Figure 16: User Study on Different K-Facet Selection Algorithms

7. RELATED WORK

In section 2 we presented a detailed comparison of FacetedPedia with a broad set of existing faceted search systems based on a taxonomy. In this section we discuss related works that are focused towards providing querying and exploring facilities on top of Wikipedia in addition to keyword search.

Various approaches have been pursued for enhancing keyword search on Wikipedia. PowerSet [4] uses natural language processing techniques to support simple questions and direct answers. It also provides aggregates information over search result Wikipedia articles. CompleteSearch proactively supports query formulation (by presenting relevant completions) and query refinement through categories (by presenting matching categories) [6]. The "facets" there refer to the following three dimensions: a display of query completions matching the query terms (as prefix); a display of category names matching the query terms (as prefix); and a display of matching categories (for which the result articles belong to). Clearly these facets are very different from the typical notion of facets as dimensions or attributes, which is our focus in this paper.

Several works explicitly support structured queries on Wikipedia. DBPedia [5] allows users to ask expressive queries against structured information extracted from Wikipedia, including infoboxes, categories, images, geo-coordinates and hyperlinks. In [8] relational tables are used to store information extraction results from Wikipedia and thus SQL-style queries can be issued over the extracted information. In [29, 24] the authors studied how to rank entities returned as answers to a keyword query.

YAGO [23] is a semantic knowledge base built from Wikipedia, allowing semantic queries over the knowledge base. Semantic Wikipedia [25] provides an extension to Wikipedia that enables users to manually specify the types of links between articles and the types of data values inside articles when they are authoring the articles. [26] built machine learning systems to automatically create and enhance several types of structures in Wikipedia, including infoboxes and link structures. Such manually or automatically generated structured and semantic information could be useful in the creation of faceted interfaces since they explicitly provide the attributes of articles as well as the relationships between articles.

8. CONCLUSION

In this paper we proposed FacetedPedia, a faceted retrieval system for information discovery and exploration over Wikipedia. This system provides a dynamic and automated faceted search interface for users to browse and navigate articles that are retrieved as a result of a keyword query. The interface consists of multiple facets, with a hierarchy of Wikipedia categories on each facet. Given the sheer size and complexity of Wikipedia, we propose metrics for ranking multi-facet interfaces as well as efficient algorithms to compute them. Our experimental evaluation and user study verify the effectiveness of our methods in generating useful faceted interfaces.

Our work poses several open problems for the future. One of the main tasks ahead of us is to investigate whether our faceted interface framework applies to other datasets besides Wikipedia, especially datasets that contain intensive hyperlinks and folksonomies created by collaborative users. It may even be possible to scale out our approaches to the Web - i.e., to offer to dynamic and automated faceted search facilities for the Web.

9. REFERENCES

[1] http://en.wikipedia.org/wiki/category:fundamental.

[2] http://en.wikipedia.org/wiki/wikipedia.

[3] http://www.alexa.com.

[4] http://www.powerset.com.

[5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, , and Z. Ives. DBpedia: A nucleus for a web of open data. In *6th Int.l Semantic Web Conf.*, 2007.

[6] H. Bast and I. Weber. The completesearch engine: Interactive, efficient, and towards IR & DB integration. In *CIDR*, pages 88–95, 2007.

[7] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev. Beyond basic faceted search. In *WSDM*, pages 33–44, 2008.

[8] E. Chu, A. Baid, T. Chen, A. Doan, and J. Naughton. A relational approach to incrementally extracting and querying structure in unstructured data. In *VLDB*, pages 1045–1056, 2007.

[9] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92*, pages 318–329, 1992.

[10] W. Dakka and P. Ipeirotis. Automatic extraction of useful facet hierarchies from text databases. *ICDE*, pages 466–475, 2008.

[11] W. Dakka, P. G. Ipeirotis, and K. R. Wood. Automatic construction of multifaceted browsing interfaces. In *CIKM*, pages 768–775, 2005.

[12] D. Debabrata, R. Jun, N. Megiddo, A. Ailamaki, and G. Lohman. Dynamic faceted search for discovery-driven analysis. In *CIKM*, 2008.

[13] J. Diederich and W.-T. Balke. FacetedDBLP - navigational access for digital libraries. *Bulletin of IEEE Technical Committee on Digital Libraries*, 4, Spring 2008.

[14] M. A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49(4):59–61, 2006.

[15] M. Käki. Findex: search result categories help users when document ranking fails. In *CHI '05*, pages 131–140, 2005.

[16] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[17] A. S. Pollitt. The key role of classification and indexing in view-based searching. In *IFLA*, 1997.

[18] W. Pratt, M. A. Hearst, and L. M. Fagan. A knowledge-based approach to organizing retrieved documents. In *AAAI '99/IAAI '99*, pages 80–85, 1999.

[19] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *CHI*, pages 190–197, 2001.

[20] K. A. Ross and A. Janevski. Querying faceted databases. In *the Second Workshop on Semantic Web and Databases*, 2004.

[21] S. B. Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania. Minimum effort driven dynamic faceted search in structured databases. In *CIKM*, 2008.

[22] E. Stoica, M. A. Hearst, and M. Richardson. Automating creation of hierarchical faceted metadata structures. In *Proc. NAACL-HLT 2007*, pages 244–251, 2007.

[23] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In *WWW*, pages 697–706, 2007.

[24] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in Wikipedia. In *SAC*, pages 1101–1106, 2008.

[25] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *WWW*, pages 585–594, 2006.

[26] F. Wu and D. S. Weld. Autonomously semantifying Wikipedia. In *CIKM*, pages 41–50, 2007.

[27] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *CHI '03*, pages 401–408, 2003.

[28] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. In *WWW*, pages 1361–1374, 1999.

[29] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on Wikipedia. In *CIKM*, pages 1015–1018, 2007.