Faceted Wikipedia

Chengkai Li, Ning Yan, Senjuti Basu Roy, Lekhendro Lisham, Gautam Das
Department of Computer Science and Engineering
University of Texas at Arlington
cli@uta.edu,{ning.yan,senjuti.basuroy,lisham.singh}@mavs.uta.edu,gdas@uta.edu

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 discovers a faceted interface for navigating the articles. To the best of our knowledge, Facetedpedia is the first such system for Wikipedia. Compared with other faceted retrieval systems, Facetedpedia is fully automatic and dynamic in both facets discovery and hierarchy construction, and the facets are based on rich semantic information. The essence of our approach is to build upon the collaborative vocabulary in Wikipedia, more specifically the intensive structures (hyperlinks) and folksonomy (category system). 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 faceted interface 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.

1. INTRODUCTION

Wikipedia has gained enormous popularity since its birth. It is among the top 10 popular Websites in terms of user traffic [3]. With the 2.6 million English articles by far, 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, as we often want to discover and explore "entities" that fascinate us. Given an information exploratory task, with only keyword search, one would have to digest the list of search result articles, follow hyperlinks to connected articles, adapt and perform multiple searches, and synthesize information manually. This procedure is often time-consuming and error-prone.

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* (dimension, attribute, property) 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 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.

1.1 Motivations

In this paper we propose Facetedpedia, a faceted retrieval system over Wikipedia. To illustrate its promise, we give below a motivating example of Facetedpedia. The objective of this work is to develop the framework and methods for automatic and dynamic discovery of faceted interfaces in Facetedpedia. The methods could be useful for similar information discovery tasks in other scenarios beyond Wikipedia.

Example 1 (Motivating Example): Imagine that a graduate student, Amy, doing her PhD in Theoretical Computer Science, is exploring information about renowned graph theoreticians. Impressed by the rich 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, keyword search in itself is unsuitable for such exploratory tasks. Amy would have to manually sieve through many articles, and identify, remember and synthesize relevant facts/relationships. Alternatively, Amy can try browsing the category system available in Wikipedia which is roughly a single category hierarchy, where she can start from the root and follow relevant categories and subcategories to reach various articles of interest. [1] However, the vast size and complexity of the category system would force Amy to evaluate 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 articles retrieved in response to Amy's initial keyword query), with the category hierarchy of each facet describing a dif-

[1] A Wikipedia article may belong to one or more categories. These categories are listed at the bottom of the article.

VLDB '09, August 24-28, 2009, Lyon, France

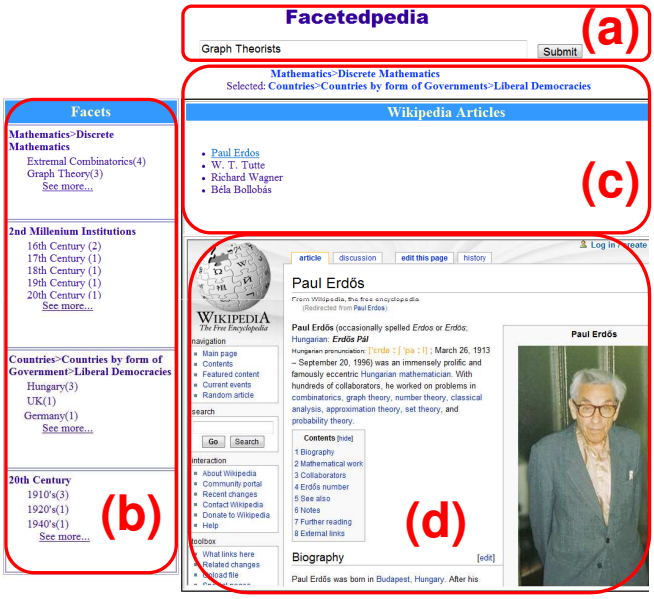


Figure 1: The graphical user interface based on facets.

ferent "dimension" of the results, would be very helpful to Amy. For example, related to "Graph Theorists", these dimensions can include *Field*, *Country*, *Institution*, *Year of Birth*, and so on. Each facet is associated with a hierarchy of categories. Given the facet hierarchies, each article can be assigned to many nodes in these hierarchies, with each assignment representing an attribute value of the article. Thus the `Wikipedia` article on graph theoretician Paul Erdös [2] might be described by the following attribute values, which are all real `Wikipedia` categories.

- Mathematics> Discrete Mathematics> Graph Theory> Graph Theorists
- Mathematics> Number Theory> Number Theorists
- Countries> Countries by Form of Government> Liberal Democracies> Hungary
- Universities and Colleges by State> New Jersey> Princeton University
- 20th Century> 1910's> 1913

For example, the fourth line above is a path of categories in the hierarchy associated with the facet *Institution*. The path indicates that Paul Erdös has a relationship with Princeton University, which is in New Jersey, and so on. Note that the category "Princeton University" is not directly assigned to "Paul Erdös" in `Wikipedia`. However, the system could possibly identify this relationship if it realizes that the article "Paul Erdös" contains a hyperlink to the `Wikipedia` article "Princeton University".

The hypothetical interface of `Facetedpedia` is shown in Figure 1. The system takes a keyword search query from Amy 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)) for the top s articles in this ranked list. Amy can navigate up and down on the multiple facets to easily explore the covered articles. On each facet, the current category path that Amy has selected is shown on the top, followed by the available subcategories following the current category path, allowing easy navigation by Amy. The interface also shows the titles of the covered articles (region (c)), i.e., those articles that belong to all the categories that Amy has selected on all the facets. When Amy clicks one title, the corresponding `Wikipedia` article would be shown (region (d)). ■

Faceted interface is an idea for user interfaces that organizes and groups retrieval results. It has become influential over the last few years and we have seen an explosive growth of interests in its application [16, 26, 14, 26, 14, 21, 11, 10, 19, 20, 13, 12, 7]. Commercial faceted search systems have been adopted by E-commerce vendors (such as `Endeca`, `IBM`, and `Mercado`), as well as several E-commerce Websites (e.g., `eBay.com`, `Amazon.com`). The utility of faceted interfaces is investigated in various studies [16, 14, 17, 26, 15, 17, 18, 14], where it has been shown users engaged in exploratory tasks often prefer such result groupings over simple ranked result-lists (commonly provided by search engines), as well as over competing ideas that also organize retrieval results, such as clustering and classification [9, 27, 15]. For example, users often do not like the disorderly groupings produced by clustering methods, preferring the more understandable category hierarchies of faceted interfaces.

Faceted interfaces over `Wikipedia` could be especially useful, due to the distinguishing features of `Wikipedia` articles that separate them from common Web pages. With one-to-one mapping to real-world entities (virtual or physical) `Wikipedia` articles represent a huge repository of human knowledge of the world. It is thus natural for users to perform discovery and exploratory of such entities. Moreover, in contrast to generic Web pages, `Wikipedia` articles are associated with rich vocabulary created by users collaboratively representing collective intelligence and semantic information. To name just a few, examples of such information 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. Therefore `Wikipedia` is a suitable subject for more sophisticated access mechanisms, such as faceted interface, in addition to keyword search, and at the same time makes automation of such mechanisms more achievable than on general Web pages.

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 navigating through the multiple 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 content 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 schema-available objects, 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 on static facets would be futile.

`Facetedpedia` is a significant and challenging research undertaking. Even the notion of "facet" itself does not arise automatically in `Wikipedia`, leave alone discovering a faceted interface. 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

[2] `http://en.wikipedia.org/wiki/Paul_Erd%C5%91s`

the dimensions or attributes of a Wikipedia article? Where does the category hierarchy on such a dimension come from?

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

The philosophy of our approach is to exploit *collaborative vocabulary* as the backbone of faceted interfaces. Wikipedia is lack of predefined schemata and controlled vocabulary that would otherwise readily provide the facet dimensions and category hierarchies. However, the collaborative vocabulary in Wikipedia, such as the hyperlinks and the category system, represents collective intelligence of users and rich semantic information, and thus constitutes the promising basis for faceted interfaces. To be more specific, with regard to the concept of dimensions or attributes, the Wikipedia articles hyperlinked from a search result article are exploited as its attributes, as shown in Example 1. This view largely enriches the information associated with the result articles. 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.

Although the general methodology of exploiting collaborative vocabulary is intuitive and promising, it is non-trivial to discover the faceted interfaces automatically and dynamically. Given a set of query result articles, there is a large number of attribute articles which in turn have many categories associated with complex hierarchical relationships. From such an overwhelmingly large search space, we need to find a set of "good" facets, i.e., a set of category hierarchies. 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.

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

We propose a principled cost metric for ranking faceted interfaces, based on users' navigational cost in exploring the facets. We design the cost function of an individual facet by averaging over the navigational paths. We then propose an interesting idea of "integrated" facet for measuring the cost of multiple facets when they interact with each other. We investigate several important issues in applying the cost metric, including how to deal with target articles unreachable from a facet, and how the overlap between facets and data correlation affect the utility of a faceted interface.

In discovering a faceted interface, it is infeasible to directly apply the ranking functions exhaustively on all possible choices, due to the prohibitively large search space. Given the sheer size and complexity of the Wikipedia category system and hyperlink structures, there is a huge number of candidate facets that could be considered in generating a faceted interface. Furthermore, the interactions between facets make the computation of the exact cost of a faceted interface intractable. Even computing the costs of individual facets without considering the interactions is not a trivial task, given the size and complexity of Wikipedia.

Challenge 3: We must design effective and efficient faceted interface discovery algorithms based on the ranking criteria.

Our solution hinges on shrinking the search space and developing efficient algorithms in searching the space. To reduce the space of candidate facets, we focus on a subset of "safe reaching facets" that do not contain categories that cannot reach any result articles. To reduce the extremely large space of faceted interfaces, we develop a two-stage method. In the first stage we design a recursive algorithm that calculates the individual ranking scores of all the candidate facets by just one depth-first search of the relevant category hierarchy. In searching for a good faceted interface, the second stage only considers the top ranked facets that are not subsumed by other top facets. Instead of exhaustively measuring the costs of all possible faceted interfaces, we apply several heuristic-based methods including a hill climbing algorithm to look for a local optimum. To further address the challenges of capturing the interactions of multiple facets in measuring the cost of a facet interface, the hill climbing algorithm optimizes for a combination of average navigational cost and pair-wise similarity of the facets.

An important hallmark of our efforts in addressing the above challenges has been to strive for principled solutions built on strong conceptual foundations, wherever possible. In fact, a significant part of our contributions has been in the development of clean data models and precise problem formulations, as well as effective algorithms for solving them. Of course, we emphasize that Wikipedia is a large and noisy real-world dataset; thus developing a faceted search interface system that is principled yet practically effective has not always been easy. We have, at times, had to resort to using simplifying assumptions and heuristics, but have tried keep the use of such techniques to a minimum.

1.3 Summary of Contributions and Outline

In summary, this paper makes the following contributions:

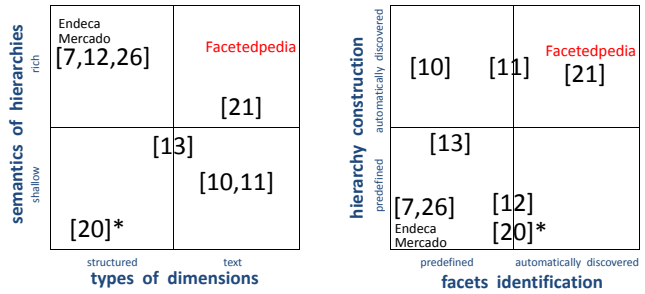
- **Concept: Faceted Wikipedia.** We propose an automatic and dynamic faceted retrieval system over Wikipedia. To the best of our knowledge, this is the first system of its kind. The key philosophy of our approach is to exploit collaborative vocabulary as the backbone of faceted interfaces. (Section 3)
- **Metrics: Facets Ranking**. Based on a user navigation model, we propose ranking metrics for measuring the "goodness" of facets, both individually and collectively. (Section 4)
- **Algorithms: Faceted Interface Discovery**. We develop effective and efficient algorithms for discovering faceted interfaces based on the ranking functions. (Section 5)
- **System Evaluation: Facetedpedia**. We empirically evaluated the system and conducted user study to compare with alternative approaches. (Section 6)

The rest of the paper is organized as follows. In Section 2 we present a comparative study of faceted retrieval systems. Section 3 formally defines the concepts and the faceted interface discovery problem. We develop the facets ranking metrics in Section 4 and faceted interface discovery algorithms in Section 5. Section 6 discusses the results of user study and experimental evaluation. Section 7 concludes the paper.

2. EXISTING FACETED RETRIEVAL SYSTEMS: A COMPARATIVE STUDY

Existing research prototypes or commercial systems mostly cannot be applied to meet our goals, because they either are based on manual or static facets construction, or are for structured records or text collections with clearly defined metadata by controlled vocabulary. Many of them in fact focus on user-interface issues or applications, instead of the discovery of faceted interfaces. Very few have investigated the problem of automatic and dynamic faceted interface discovery and none has provided faceted interface over Wikipedia. (CompleteSearch [6] supports a very special type of "facets" over Wikipedia and is discussed in Section 2.2.)

In this section we present taxonomies to characterize the relevant faceted retrieval systems and to compare them with Facetedpedia.



(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.

2.1 Taxonomies of Faceted Retrieval Systems

Figure 2(a): Taxonomy by Dimension Types and Semantics

`Facetedpedia` sits between two extreme ends of existing systems. On the one hand, we build on top of the abundant collaborative vocabulary in absence of predefined schemata; On the other hand, such collaborative vocabulary does not directly provide the facet dimensions and hierarchies for us, thus presenting a significant challenge in faceted interface discovery.

In some systems the facets are on relational data (e.g., `Endeca`, `Mercado`, [20]) or structured attributes in expert-designed schemata (e.g., [26, 12, 7]) and the hierarchies on attribute values are predefined based on domain-specific taxonomies. Since the dimensions are predefined, the hierarchies could even be manually created, therefore could have rich semantic information. In other systems a facet is a group of textual terms, over which the hierarchy is built upon thesaurus-based IS-A relationships (e.g., [21]) or frequency-based subsumption relationships between general and specific terms (e.g., [11, 10]). In such a scenario, the systems cannot leverage much semantic information. [13] is in the middle of Figure 2(a) since most of the dimensions are structured, except one special topic taxonomy on the subsumption relationships between topic words. In `Facetedpedia`, the type of attribute values is text, i.e., the titles of `Wikipedia` categories. However, `Facetedpedia` is based on `Wikipedia` folksonomy (instead of IS-A or subsumption relationships), and therefore incorporates collaboratively generated semantic information into the facet hierarchies.

Figure 2(b): Taxonomy 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. On this aspect, none of the current faceted retrieval systems could be effectively applied for `Facetedpedia`. First, there are no existing algorithms for dynamically discovering query-dependent facets. Second, most algorithms are not fully automatic in both facets identification and hierarchy construction.

In some commercial products and research systems (e.g., `Endeca`, `Mercado`, [20, 7, 26, 12]) the dimensions (such as attributes in relational tables) and hierarchies are predefined, therefore they do not discover the facets or construct the hierarchy. (In [12, 20] they do automatically select a subset of most interesting/important dimensions from the predefined ones. Moreover, [20] does not support category hierarchy on attributes.) In [11, 10] the set of facets are predefined, but the hierarchies are automatically created based on subsumption. [11] also selects the most important portion of the hierarchy for displaying in an limited space. In Diederich et al. [13] only one facet (a topic taxonomy) is automatically generated and the rest are predefined.

With respect to the automation of faceted interface discovery, the closest work to ours is the `Castanet` algorithm [21]. It automatically creates facets from the textual descriptions of a collection of items (e.g., recipes). The hierarchies for the multiple facets are obtained 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.

2.2 Other Related Work on Querying and Exploring Wikipedia

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. `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; a display of category names matching the query terms; and a display of matching categories of result articles. Clearly these facets are different from the notion of facets as dimensions or attributes, which is our focus.

Several works explicitly support structured queries on `Wikipedia`. `DBPedia` [5] allows users to ask expressive queries against structured information extracted from `Wikipedia`. [8] uses relational tables to support `SQL`-style queries over the extracted information. [28, 23] studied how to rank resulting entities of keyword queries.

`YAGO` [22] supports semantic queries over a knowledge base on `Wikipedia`. `Semantic Wikipedia` [24] extends `Wikipedia` to allow users to manually specify the types of hyperlinks and data values in articles. [25] automatically creates and enhances various structures in `Wikipedia`, including infoboxes and link structures. Such manually or automatically generated information could be useful in creating faceted interfaces since they explicitly provide the attributes of articles and the relationships between articles.

3. FACETED WIKIPEDIA BY COLLABORATIVE VOCABULARY

The goal of `Facetedpedia` is to discover faceted interfaces for exploring the dynamic keyword query results over `Wikipedia`. The fundamental philosophy of our approach is to exploit *collaborative vocabulary* as the backbone of faceted interfaces. `Wikipedia` articles are different from both structured data such as relational tables and unstructured data such as generic Web pages and text document collections. On the one hand, predefined schemata and controlled vocabulary such as taxonomy are missing in `Wikipedia`, therefore facet dimensions and their category hierarchies are not readily available. On the other hand, `Wikipedia` features user-generated collaborative vocabulary, including the "grassroots" taxonomy such as the category system and schema-like information such as the "infoboxes". [3] Even hyperlinks in this environment is an instance of collaborative vocabulary in a broader sense, as they indicate users' collaborative endorsement of relationships between entities. Such collaborative vocabulary represents collective intelligence of users and rich semantic information, and thus constitutes the promising basis for faceted interfaces.

[3] An infobox is a summary table on a `Wikipedia` article.

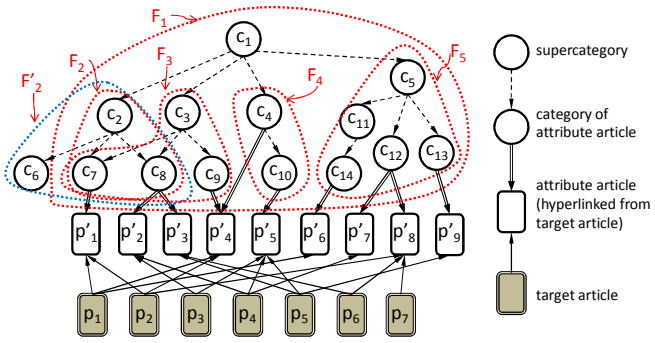


Figure 3: The concept of facet.

With regard to **the concept of dimension**, the `Wikipedia` articles hyperlinked from a search result article are exploited as its attributes, as shown in Example 1. The underlying intuition is simple. The fact that the authors of an article collaboratively 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, "Princeton University" becomes an attribute of "Paul Erdös". During exploration, the categories of "Princeton University" are helpful in reaching "Paul Erdös" even though "Paul Erdös" itself does not belong to those categories. The user could then navigate to "Paul Erdös" by specifying (via the facet on *Institution*) that she is looking for a graph theoretician that has relationships with some university in New Jersey.

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".

Based on the intuition above, we now formally define the concepts in our framework and present the specification of the faceted interface discovery problem.

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 $p_j \in \mathcal{T}$. [4] ■

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). [5] All the categories of articles are direct or indirect subcategories of the root, `Category:Fundamental` [1]. [6] [7]

Definition 2 (Category Hierarchy): The `Wikipedia` category hierarchy is a connected, rooted directed acyclic graph $\mathcal{H}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$, 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 $\mathcal{H}$, $r_{\mathcal{H}}$, is `Category:Fundamental`. ■

We define *facet* as follows.

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 $\mathcal{H}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$, 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 $\mathcal{F}$. ■

Example 2 (Running Example): Figure 3 illustrates the basic concepts. There are 7 target articles ($p_1, \ldots, p_7$) and 9 attribute articles ($p'_1, \ldots, p'_9$). The category hierarchy has 14 categories ($c_1, \ldots, c_{14}$). The figure highlights 6 facets ($\mathcal{F}_1, \ldots, \mathcal{F}_5$, and $\mathcal{F}'_2$). For instance, F_2 is rooted at c_2 and consists of 3 categories (c_2, c_7, c_8) and 2 edges ($c_2 \dashrightarrow c_7$, $c_2 \dashrightarrow c_8$). There are many more facets since every rooted and connected subgraph of the hierarchy is a facet. Note that the figure may give the impression that edges such as $c_5 \dashrightarrow c_{12}$ and $c_7 \Rightarrow p'_1$ are unnecessary since there is only one choice under c_5 and c_7, respectively. The example is small due to space limitations. Such single outgoing edge is very rare in the real `Wikipedia` category hierarchy. We will use Figure 3 as the running example throughout the paper. ■

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 $c_1 \dashrightarrow \ldots \dashrightarrow c_t$, 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 $c_1 \dashrightarrow \ldots \dashrightarrow c_t \Rightarrow p' \leftarrow p$, where,

- $c_1 \dashrightarrow \ldots \dashrightarrow c_t$ 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 $c_1 \dashrightarrow \ldots \dashrightarrow c_t \Rightarrow p' \leftarrow p$, we say that the corresponding category path $c_1 \dashrightarrow \ldots \dashrightarrow c_t$ *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 $1 \le i \le t$). ■

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

[5] 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.

[6] An alternative root is `Main Topic Classification`, which has more detailed initial subcategories than `Fundamental`.

[7] We only consider categories of articles. The root of the whole category system, `Contents`, contains categories of all types.

Note that in Definition 5 $p' \leftarrow p$ 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'.

Example 3 (Category Path, Navigational Path): Continue the running example. In Figure 3, two examples of navigational paths are $c_2 \dashrightarrow c_8 \Rightarrow p'_3 \leftarrow p_5$ and $c_5 \dashrightarrow c_{13} \Rightarrow p'_9 \leftarrow p_5$. ■

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 "dead end" 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 "dead end", 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.

By Definition 6, 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 is thus 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 $\mathcal{T}$. ■

Example 4 (Faceted Interface): Continue the running example. In Figure 3, an example 2-facet interface is $I = \{\mathcal{F}_2, \mathcal{F}_5\}$. However, $\{\mathcal{F}'_2, \mathcal{F}_5\}$ is not a valid faceted interface because $\mathcal{F}'_2$ is not a safe reaching facet, as category c_6 cannot reach any target articles. ■

Based on the formal definitions, the **Faceted Interface Discovery Problem** is: Given the category hierarchy $\mathcal{H}(r_\mathcal{H}, \mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$, for a keyword query q and its resulting target articles $\mathcal{T}$ and corresponding attribute articles $\mathcal{A}$, find the "best" faceted interface with k facets. We shall develop the notion of "best" in Section 4.

4. FACETS RANKING METRICS

The search space of the faceted interface discovery problem is prohibitively large. Given $\mathcal{T}$, the set of s result `Wikipedia` articles to a keyword query, there are 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 about 2.6 million English articles with hundreds of millions of internal links. The category system $\mathcal{H}$ contains close to half a million categories and several million category-subcategory relationships. Any rooted and connected subgraph of $\mathcal{H}$ that safely reaches $\mathcal{T}$ is a candidate facet by definition, and any combination of k facets would be a candidate faceted interface. Given the large space, we need ranking metrics for measuring the "goodness" of facets, both individually and collectively as interfaces.

Given that a faceted interface is for a user to navigate through the associated category hierarchies and ultimately reaching the target articles, it is natural to rank the interfaces by the user's navigational cost, i.e., the amount of effort undertaken by the user during navigation.[8] The "best" k-facet interface is the one with the smallest cost. Therefore as the basis of such ranking metrics, we model users' navigational behaviors as follows.

User Navigation Model: A user navigates multiple facets in a k-facet interface. At the beginning, the navigation starts from the roots of all the k facets. At each step, the user picks one facet and examines the set of subcategories available at the current category on that facet. She follows one subcategory to further go down the category hierarchy. Alternatively the user may select one of the attribute articles reachable from the current category. The selections made on the k facets together form a conjunctive query. After the selection at each step, the list of target articles that satisfy the conjunctive query are brought to the user. The navigation terminates when the user decides that she has seen desirable target articles.

Example 5 (Navigation in Faceted Interface): We continue the running example. Consider a faceted interface consisting of multiple facets from Figure 3, including $\mathcal{F}_2$ and $\mathcal{F}_5$. A sequence of navigational steps on this interface are shown in Figure 4. At the beginning, the user has not selected which facets to explore, therefore all 7 target articles are available to the user (step 1). Once the user decides to explore $\mathcal{F}_2$ which starts from c_2, p_7 is filtered out since it is unreachable from $\mathcal{F}_2$ (step 2). The user then selects c_5, which further removes p_3 from consideration for the same reason (step 3). After the user further explores $\mathcal{F}_2$ by choosing c_8 (step 4), c_{11} is not a choice under c_5 anymore because no target articles could be reached by both $c_2 \dashrightarrow c_8$ and $c_5 \dashrightarrow c_{11}$. The user continues to explore $\mathcal{F}_5$ by choosing c_{13} (step 5), which removes p'_2 and also trims down the satisfactory target articles to $\{p_5\}$. The user may decide she has seen desirable articles and the navigation stops. ■

4.1 Single-Facet Ranking

In this section we focus on how to measure the costs of facets individually. Based on the navigational model, we compute the navigational cost of a facet as the average cost of its navigational paths. The core of this metric thus is the measure of the cost of a navigational path. Intuitively a low-cost navigational path, i.e., a path that demands small user effort, should have a small number of steps and at each step only requires the user to browse a small number of choices. Therefore, we formally define the cost of a navigational path as the summation of the fan-outs (i.e., number of choices), in logarithmic form, [9] at every step.

Definition 8 (Cost of Navigational Path): With respect to the target articles $\mathcal{T}$, the corresponding attribute articles $\mathcal{A}$, and a facet

[8] [20] also selects facets based on navigational costs, although their system is of a different nature, as discussed in Section 2.

[9] The intuition behind the logarithmic form can be explained. When presented with a number of choices, the user does not necessarily scan through the choices linearly. For example, the user interface could be designed in such a way that the user could find the desired choice by binary search over the list of choices.

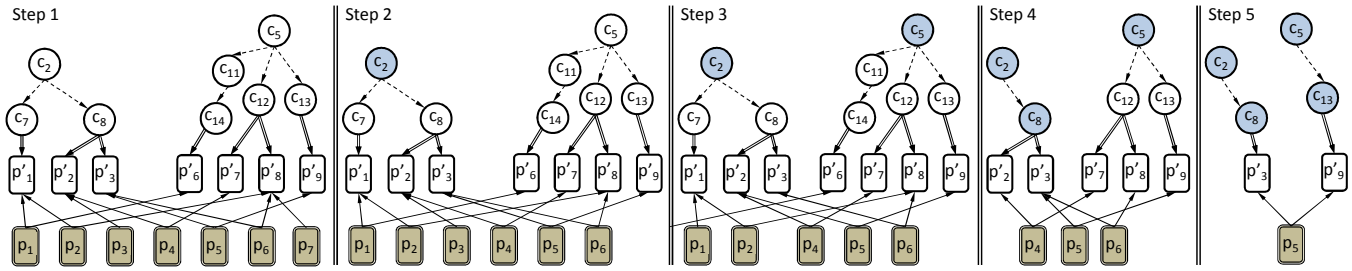


Figure 4: The navigation on two facets, $\mathcal{I} = \{\mathcal{F}_2, \mathcal{F}_5\}$.

$\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, the cost of a navigational path in $\mathcal{F}$, l=c_1 ⇢ ... ⇢ c_t⇒p'←p, is defined as

$$cost(l) = log_2(fanout(p')) + \sum_{c \in \{c_1, ..., c_t\}} log_2(fanout(c)) \quad (1)$$

In Equation 1, $fanout(p')$ is the number of (directly) reachable target articles through the attribute article p',

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

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

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 $\mathcal{F}$,

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

Note that we make several assumptions for simplicity of the model. The above cost formula only captures the "browsing" cost. A full-fledged formula would need to incorporate other costs, including the "clicking" cost associated with the selection of a choice and the cost of "backward" navigation when the user decides to go back and change a previous selection. Furthermore, we assume that the user will always complete the navigational path till reaching the target articles. However, in reality the user may stop in the middle when she already finds desirable target articles that are reachable from the current selection of category. We leave the investigation of more sophisticated models to future study.

Example 6 (Cost of Navigational Path): We continue the running example. Given l=c_5 ⇢ c_{12}⇒p'_8←p_6, a navigational path of $\mathcal{F}_5$ in Figure 3, the cost of l is $cost(l)$=$fanout(c_5)$+$fanout(c_{12})$+$fanout(p'_8)$=$log_2(3)$+$log_2(2)$+$log_2(3)$=4.17. ■

Albeit the basis of our facets ranking metrics, navigational cost only is not sufficient in measuring the goodness of a facet. More specifically we need to take into consideration the scenario when a facet cannot fully reach all the target articles. In an environment with prescribed rigorous schemata, such as relational databases, 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 common that a facet cannot reach some of the target articles. Missing the unreachable articles presents an unsatisfactory user experience. The aforementioned definition of navigational cost does not consider this. In fact, low-cost and high-coverage could be two qualities that compete with each other. On the one hand, a low-cost facet in theory could be one that is very small and reaches only a small portion of the target articles. On the other hand, a comprehensive facet with high coverage may tend to be wider and deeper, thus more costly. Therefore we must incorporate into the cost formula the notion of "coverage", i.e., the ability of a facet to reach as many target articles as possible.

In ranking the facets, to combine navigational cost with coverage, we penalize the facet by associating an expensive pseudo path with each unreachable article, and combine the pseudo cost with the cost of regular navigational paths. Formally, we define the cost of a facet as the average cost in reaching each target article based on the assumption that they are all equally important (since they are the top ranked results to a keyword search).

Definition 9 (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⇢...⇒p'←p, where p' is an attribute article of p. The reachable target articles from $\mathcal{F}$ is $\mathcal{T}_r$=$\{p | p \in \mathcal{T} \wedge p$ 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 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 the average cost in reaching each target article. The cost for a reachable target article is the average cost of navigational paths that start from the root r and reach the target, and the cost for an unreachable target is a pseudo cost $penalty$.

$$cost(\mathcal{F}_r) = \frac{1}{|\mathcal{T}|} \times (\sum_{p \in \mathcal{T}_r} cost(\mathcal{F}_r, p) + penalty \times |\mathcal{T} - \mathcal{T}_r|) \quad (7)$$

where $cost(\mathcal{F}_r, p)$ is the average cost of reaching p from r,

$$cost(\mathcal{F}_r, p) = \frac{\sum_{l \in l_p} cost(l)}{|l_p|} \quad (8)$$

where l_p is the set of navigational paths in $\mathcal{F}$ that reach p from r,

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

In Equation 7, the parameter $penalty$ is the cost of an expensive pseudo path that "reaches" the unreachable target articles, i.e., $\mathcal{T} - \mathcal{T}_r$, for penalizing a facet for not reaching such articles. Its value is empirically selected and shall be discussed in Section 6.

Example 7 (Cost of Facet): We continue the running example. Figure 5 shows the costs of the 5 highlighted facets in Figure 3, together with their category hierarchies and reachable attribute and target articles. It does not show $\mathcal{F}_1$ which is Figure 3 itself excluding c_6. The cost is obtained by Formula 7, assuming $penalty$=7. For instance, $cost(\mathcal{F}_2)$=$\frac{1}{7}$×($\sum_{p \in \{p_1, p_2, p_3, p_4, p_5, p_6\}}$ $cost(\mathcal{F}_2, p)$ + $penalty$×$|\mathcal{T} - \mathcal{T}_{\mathcal{F}_2}|$) = $\frac{1}{7}$× (16 +7 × 1)=3.286. The costs of other facets are computed in the same way. As the figure shows, $\mathcal{F}_2$ and $\mathcal{F}_5$ achieve lower costs than other facets. Even though the paths in $\mathcal{F}_4$ are cheap, $\mathcal{F}_4$ has higher cost due to penalty for unreachable target articles (p_6 and p_7). $\mathcal{F}_1$ is even more costly due to its wider and deeper hierarchy, although it reaches all target articles. ■

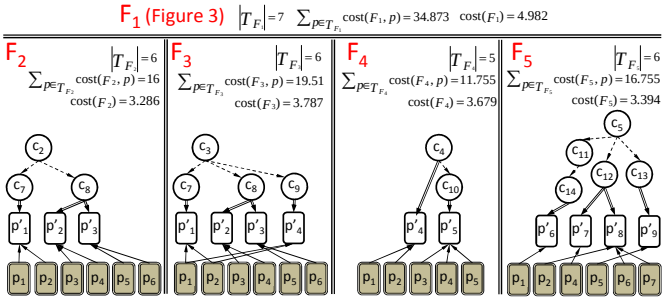


Figure 5: Navigational costs of facets.

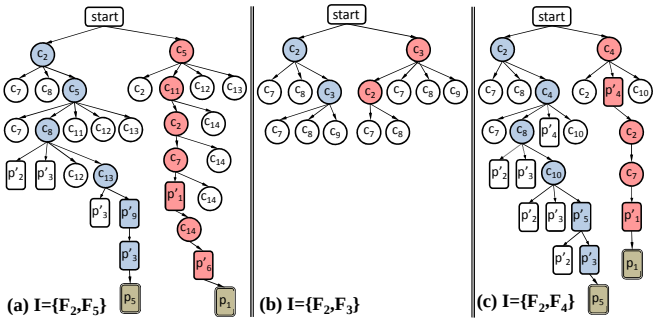


Figure 6: The sequences of navigational steps.

4.2 Multi-Facet Ranking

Even provided with the cost metrics of individual facets in Section 4.1, measuring the "goodness" of a faceted interface, i.e., a set of facets, is not straightforward. This is because the best k-facet interface may not be simply the top-k facets ranked by the single-facet cost function. The reason is that when the user navigates multiple facets, the selection made at one facet has impact on the choices available on other facets, as illustrated by Example 5.

In order to directly follow the approach of ranking facets by navigational cost as in Section 4.1, in principle we could represent the navigation steps on multiple facets as if the navigation is on one **"integrated" facet**. To illustrate, consider the navigation on a 2-facet interface $\mathcal{I}=\{\mathcal{F}_2, \mathcal{F}_5\}$ from Figure 3. Two possible sequences of navigational steps on this interface are shown in Figure 6(a). One is c_2, c_5, c_8, c_{13}, p'_9, p'_3, p_5, which is the steps taken by the user in Figure 4, followed by choosing p'_9, p'_3, and finally p_5. (Remember, for simplification of the model, we assumed that the user will always complete navigational paths till reaching the target articles.) At each step, the available choices from both facets are put together as the choices in the "integrated" facet. Note that after c_8 is chosen, c_{12} and c_{13} are still valid choices but c_{11} is not available anymore because c_{11} cannot reach the target articles that c_8 reaches. For the same reason, after c_{13} is chosen, p'_3 is still a valid choice but p'_2 is not available anymore. The other highlighted sequence of steps is c_5, c_{11}, c_2, c_7, p'_1, c_{14}, p'_6, p_1. There are many more possible sequences not shown in the figure due to space limitations.

With the concept of "integrated" facet, one may immediately apply Definition 10 to define the cost of faceted interface. However, direct cost computation by this definition is infeasible because that entails exhaustively pre-computing all possible sequences of interleaving navigational steps across all the facets in the interface. The interaction between facets is query- and data-dependent, rendering the exhaustive computation practically impossible.

However, the "integrated" facet does shed light on what are the characteristics of good faceted interfaces. In general an interface should not include two facets whose structures overlap much. Imagine a special case where two facets form a subsumption relationship, i.e., the root category of one facet is a supercategory of the root of the other facet. Presenting both facets would not be desirable since they overlap significantly, thus cannot capture the properties of reaching target articles through multiple dimensions. As a more concrete example, consider the navigational steps of $\mathcal{F}_2$ and $\mathcal{F}_3$ in Figure 6(b). After the user selects c_2 from $\mathcal{F}_2$ and then c_3 from $\mathcal{F}_3$, the navigation degenerates into exploring the single facet $\mathcal{F}_3$ itself. Similarly it degenerates into $\mathcal{F}_2$ if the user selects c_3 and then c_2. (Thus we only show the two steps of selecting c_2 and c_3.) Having the two facets does not have any advantages over having just any of the two.

Therefore, in order to make our task of computing the best k-facet interface more tractable, we propose to measure the goodness of a k-facet interface by the *average pair-wise similarity* of the k facets, in addition to their average cost. The similarity of two facets is the degree of overlap in their category hierarchies and associated attribute articles. In Section 5.3 we discuss how to incorporate the similarity measure with the navigational cost. Below we give the definition of the similarity function.

Definition 11 (Average Similarity of k-Facet Interface): The average pair-wise similarity of a k-facet interface is defined as

$$sim(\mathcal{I}=\{\mathcal{F}_1,...,\mathcal{F}_k\}) = \frac{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} sim(\mathcal{F}_i,\mathcal{F}_j)}{k(k-1)/2}, \quad (10)$$

where $sim(\mathcal{F}_i,\mathcal{F}_j)$ is defined by the Jaccard coefficient,

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

where $\mathcal{C}_{\mathcal{F}_i}$ is the set of categories in $\mathcal{F}_i$ (Definition 3) and $\mathcal{A}_{\mathcal{F}_i}$ is the set of attribute articles reachable from $\mathcal{F}_i$,

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

Example 8 (Similarity of Facets): Continue the running example. Consider the 5 highlighted facets in Figure 3. The similarity between $\mathcal{F}_2$ and $\mathcal{F}_3$ is $sim(\mathcal{F}_2,\mathcal{F}_3)=\frac{|\mathcal{C}_{\mathcal{F}_2} \bigcap \mathcal{C}_{\mathcal{F}_3}|+|\mathcal{A}_{\mathcal{F}_2} \bigcap \mathcal{A}_{\mathcal{F}_3}|}{|\mathcal{C}_{\mathcal{F}_2} \bigcup \mathcal{C}_{\mathcal{F}_3}|+|\mathcal{A}_{\mathcal{F}_2} \bigcup \mathcal{A}_{\mathcal{F}_3}|} = \frac{|\{c_7,c_8\}|+|\{p'_1,p'_2,p'_3\}|}{|\{c_2,c_7,c_8,c_3,c_9\}|+|\{p'_1,p'_2,p'_3,p'_4\}|}=5/9$. Similarly other pair-wise similarities are $sim(\mathcal{F}_1,\mathcal{F}_2)=6/22$, $sim(\mathcal{F}_1,\mathcal{F}_3)=7/22$, $sim(\mathcal{F}_1,\mathcal{F}_4)=4/22$, $sim(\mathcal{F}_1,\mathcal{F}_5)=9/22$, $sim(\mathcal{F}_2,\mathcal{F}_4)=0$, $sim(\mathcal{F}_2,\mathcal{F}_5)=0$, $sim(\mathcal{F}_3,\mathcal{F}_4)=1/11$, $sim(\mathcal{F}_3,\mathcal{F}_5)=0$, $sim(\mathcal{F}_4,\mathcal{F}_5)=0$. Given a faceted interface $\mathcal{I}=\{\mathcal{F}_2,\mathcal{F}_3,\mathcal{F}_5\}$, the average pari-wise similarity is $sim(\mathcal{I})=(sim(\mathcal{F}_2,\mathcal{F}_3)+sim(\mathcal{F}_2,\mathcal{F}_5)+sim(\mathcal{F}_3,\mathcal{F}_5))/3=5/27$. ■

We should note that the notion of similarity or overlap does not always fully capture the goodness of a faceted interface. In addition to overlap, another aspect that has impacts on the interactions of multiple facets is *data correlation*. As an example, observe $\mathcal{F}_2$ and $\mathcal{F}_4$ in Figure 5. Although $sim(\mathcal{F}_2,\mathcal{F}_4)=0$, having them both in a faceted interface is not beneficial. The two facets are highly correlated, because $\{p_1,p_2\}$ are reachable only from c_7 in $\mathcal{F}_2$ and p'_4 in $\mathcal{F}_4$ and $\{p_3,p_4,p_5\}$ are reachable only from c_8 in $\mathcal{F}_2$ and c_{10} in $\mathcal{F}_4$. Consider the left sequence of navigational steps in Figure 6(c). Once the user selects c_2, c_4, and c_8, further selections on $\mathcal{F}_4$ does not help to filter the satisfying result articles. p'_4 is not an choice on $\mathcal{F}_4$ anymore since p_1 and p_2 have been excluded from consideration after c_8 is chosen. Selecting c_{10} and p'_5 is a waste of time because that still leaves p'_2 and p'_3 as the choices available on $\mathcal{F}_2$ and $\{p_3,p_4,p_5\}$ as the satisfying target articles. Similarly the right sequence of steps shows that selecting anything on $\mathcal{F}_2$ is not helpful after p'_4 is chosen on $\mathcal{F}_4$, for the same reason.

Addressing the issue of data correlation is a very challenging problem because it is both data- and query-dependent and faceted interfaces must be generated dynamically. As a first attempt to discover faceted interfaces for `Wikipedia`, we focus on the overlap

Algorithm 1: Faceted Interface Discovery

Input: q:keyword search query; $\mathcal{H}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$: category hierarchy.

Output: $\mathcal{I}_k$: a discovered faceted interface with k facets

$\mathcal{T} \leftarrow$ the top-s ranked results (`Wikipedia` articles) for query q
Algorithm 2 // get attribute articles $\mathcal{A}$ and relevant category hierarchy $\mathcal{RCH}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{RCH}}, \mathcal{E}_{\mathcal{RCH}})$.
Algorithm 3 // rank individual facets.
Algorithm 5 // select k facets to form an interface.
return $\mathcal{I}_k$

measure of multiple facets in capturing their interactions, and we believe a more comprehensive treatment calls for the study of other aspects as well including data correlation.

5. ALGORITHMS

With the facets ranking metrics developed in Section 4, one straightforward approach of faceted interface discovery is to enumerate all possible k-facet interfaces with respect to the category hierarchy $\mathcal{H}$ and apply the ranking metrics directly to find the best interface. Such a naïve method results in the exhaustive examination of all possible combinations of k instances of all possible facets, i.e., rooted and connected subgraphs of $\mathcal{H}$. Clearly it is a prohibitively large search space, given the sheer size and complexity of `Wikipedia`. The naïve technique would be extremely costly. Therefore finding the best k-facet interface is a difficult optimization problem.

The outline of our k-facet discovery algorithm is in Algorithm 1. Our solution hinges on (a) reducing the search space; and (b) developing effective and efficient algorithms for searching the space.

Reducing the Search Space: Note that in the k-facet interface discovery problem, there are two search spaces: the space of facets and the space of k-facet interfaces, which are sets of k facets. To reduce the space of candidate facets, we focus on a subset of the safe reaching facets (Definition 6), $\mathcal{RCH}$-induced facets, which are the facets that contain all the descendant categories of their roots (Section 5.1). To reduce the extremely large space of faceted interfaces, we rank the facets individually by their navigational costs (Section 5.2) and only consider the top ranked facets that are not subsumed by other top facets (Section 5.3).

Searching the Space: In searching for a faceted interface, instead of exhaustively examining all possible interfaces to find the best one, we apply several heuristic-based methods including a hill climbing algorithm to look for a local optimum (Section 5.3). To further address the challenges of capturing the interactions of multiple facets in measuring the cost of a facet interface, the hill climbing algorithm optimizes for a combination of average navigational cost and pair-wise similarity of the facets, as motivated in Section 4.2.

5.1 Relevant Category Hierarchy (Algorithm 2)

By Definition 7, the facets in a faceted interface must be safe reaching facets, i.e., they do not contain "unnecessary" categories that cannot reach any target articles. Therefore the categories that appear in any safe reaching facet can only come from the *relevant category hierarchy* ($\mathcal{RCH}$), which is a subgraph of the `Wikipedia` category hierarchy $\mathcal{H}$, defined as follows.

Definition 12 (Relevant Category Hierarchy): With respect to the category hierarchy $\mathcal{H}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$, 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. There exists an edge (category-subcategory relationship) between two categories in $\mathcal{RCH}$ if the same edge exists in $\mathcal{H}$. By this definition the root of $\mathcal{H}$ is also the root of $\mathcal{RCH}$. ■

Algorithm 2: Construct RCH and Get Attribute Articles

Input: $\mathcal{T}$: target articles; $\mathcal{H}$: category hierarchy.

Output: $\mathcal{A}$: attribute articles; $\mathcal{RCH}$: relevant category hierarchy.

// get attribute articles.
$\mathcal{A} \leftarrow \phi$; $\mathcal{C}_{\mathcal{RCH}} \leftarrow \phi$; $\mathcal{E}_{\mathcal{RCH}} \leftarrow \phi$
foreach $p \in \mathcal{T}$ **do**
  foreach $p \rightarrow p'$, *i.e., a hyperlink from* p *to* p' **do**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{p'\}$
// start from the categories of attribute articles.
foreach $p' \in \mathcal{A}$ **do**
  foreach $c \Rightarrow p'$, *i.e., a category of* p' **do**
    $\mathcal{C}_{\mathcal{RCH}} \leftarrow \mathcal{C}_{\mathcal{RCH}} \cup \{c\}$
// recursively obtain the supercategories.
$\mathcal{C} \leftarrow \mathcal{C}_{\mathcal{RCH}}$; $\mathcal{C}' \leftarrow \phi$
while $\mathcal{C}$ *is not empty* **do**
  foreach $c \in \mathcal{C}$ **do**
    foreach $c' \dashrightarrow c \in \mathcal{E}_{\mathcal{H}}$ **do**
      $\mathcal{E}_{\mathcal{RCH}} \leftarrow \mathcal{E}_{\mathcal{RCH}} \cup \{c' \dashrightarrow c\}$
      if $c' \notin \mathcal{C}_{\mathcal{RCH}}$ **then**
        $\mathcal{C}_{\mathcal{RCH}} \leftarrow \mathcal{C}_{\mathcal{RCH}} \cup \{c'\}$; $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{c'\}$
  $\mathcal{C} \leftarrow \mathcal{C}'$; $\mathcal{C}' \leftarrow \phi$
return $\mathcal{A}$ *and* $\mathcal{RCH}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{RCH}}, \mathcal{E}_{\mathcal{RCH}})$

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 $\mathcal{H}$, it can be still much larger than we need. As an empirical evidence, we find that for 200 target articles, there could be thousands of corresponding attribute articles, and the $\mathcal{RCH}$ could contain tens of thousands of categories and even much larger number of category-subcategory relationships. Thus we further shrink the space by considering only one type of safe reaching facets, *$\mathcal{RCH}$-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., in $\mathcal{F}$ all the descendants of the root r and their category-subcategory relationships are retained from $\mathcal{RCH}$. ■

Example 9 ($\mathcal{RCH}$ and $\mathcal{RCH}$-Induced Facet): Continue the running example. In Figure 3, the $\mathcal{RCH}$ contains all the categories in the category hierarchy $\mathcal{H}$ except c_6 (and thus the edge $c_2 \dashrightarrow c_6$), since c_6 cannot reach any target articles. $\mathcal{F}_2$ is an $\mathcal{RCH}$-induced facet, while $\mathcal{F}_2'$ is not, i.e., $\mathcal{F}_2'$ is an unnecessary facet. ■

The following Theorem 1 guarantees that, when searching in $\mathcal{RCH}$-induced facets, we would only encounter safe reaching facets in $\mathcal{RCH}$. We omit the simple proof due to space limitations.

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

Note that we cannot guarantee that every safe reaching facet is $\mathcal{RCH}$-induced, i.e., some safe reaching facet may not be included

Algorithm 3: Facets Ranking

Input: $\mathcal{T}$: targets; $\mathcal{A}$: attributes; $\mathcal{RCH}$: relevant category hierarchy.

Output: $\mathcal{I}_n$: the top n $\mathcal{RCH}$-induced facets with the smallest costs.

// get reachable target articles for each attribute article.
foreach $p' \in \mathcal{A}$ **do**
  $\mathcal{T}_{p'} \leftarrow \{p | p \in \mathcal{T} \wedge \exists\, p \rightarrow p'$ (hyperlink from p to $p')\}$
  $fanout(p') \leftarrow |\mathcal{T}_{p'}|$
initialize $visited(r)$ to be $False$ for every $r \in \mathcal{C}_{\mathcal{RCH}}$.
$ComputeCost(r_{\mathcal{H}})$ // recursively compute the costs of all the $\mathcal{RCH}$-induced facets, starting from the root of $\mathcal{RCH}$, i.e., $r_{\mathcal{H}}$.
$\mathcal{I}_n \leftarrow$ the top n $\mathcal{RCH}$-induced facets with the smallest costs.
return $\mathcal{I}_n$

in this new space. By that sense this shrinking of space is lossy. Intuitively the $\mathcal{RCH}$-induced facets are those "maximal" safe reaching facets since they retain from $\mathcal{RCH}$ all the subcategories and category-subcategory relationships under the roots. It is very possible that a "good" facet is not such a "maximal" facet. We focus on the space of $\mathcal{RCH}$-induced facets and how to cover other "good" facets forms a very challenging research problem for future study.

5.2 Ranking the Facets (Algorithm 3)

Given the $\mathcal{RCH}$ with respect to a keyword search query q, we rank the $\mathcal{RCH}$-induced facets individually by their navigational costs. Only top n facets with the smallest costs are considered in searching for a faceted interface. In ranking the facets, one straightforward approach is to enumerate all the $\mathcal{RCH}$-induced facets and to separately compute the cost of each facet by enumerating all the navigational paths in it. Therefore such a straightforward approach is exponentially complex due to repeated traversal of the edges in $\mathcal{RCH}$, because the $\mathcal{RCH}$-induced facets would have many common categories and category-subcategory relationships. (Remember every category in $\mathcal{RCH}$ corresponds to the root of an $\mathcal{RCH}$-induced facet that includes all the descendant categories of the root.)

To avoid such a costly method, we design a recursive algorithm that calculates the navigational costs of all the $\mathcal{RCH}$-induced facets by only one depth-first search (DFS) of the category hierarchy of $\mathcal{RCH}$. The details are shown in Algorithm 3. The essence of the algorithm is to, during the recursive traversal of $\mathcal{RCH}$, book-keep the number of navigational paths in a facet in addition to its navigational cost. The bookkeeping is performed for each reachable target article because the cost is averaged across all the reachable articles by Definition 10. The cost of a facet rooted at r can be fully computed based on the book-keeped information of the facets rooted at r's subcategories, without accumulating the individual costs of the facets rooted at the further subcategories of r's subcategories. Therefore we can avoid the aforementioned repeated traversal in $\mathcal{RCH}$. More specifically, the lines 11-14 in Procedure $ComputeCost$ are for computing $cost(\mathcal{F}_r, p)$ in Formula 7. However, the algorithm does not compute it by a direct translation of Formula 8 and 1, i.e., enumerating all the navigational paths that reach p. Instead, line 12 gets $cost_1$, the total cost of all the navigational paths $r \Rightarrow p' \leftarrow p$, i.e., the ones that reach p without going through any other categories; line 13 computes $cost_2$, the total cost of all the navigational paths that go through other categories, by utilizing $cost(\mathcal{F}_c, p)$ and $pathcnt(\mathcal{F}_c, p)$ of the subcategories c, but not other descendants. The correctness of the algorithm is simple to verify by referring to Definition 8 and 10. Therefore we omit the formal correctness proof.

5.3 Searching for k-Facet Interface (Algorithm 5)

Algorithm 4: ComputeCost(r)

Input: r: the root of an $\mathcal{RCH}$-induced facet.

Output:
$cost(\mathcal{F}_r)$: cost of $\mathcal{F}_r$;
$cost(\mathcal{F}_r, p)$: average cost of reaching target article p through $\mathcal{F}_r$;
$pathcnt(\mathcal{F}_r, p)$: # of navigational paths reaching p through $\mathcal{F}_r$;
$\mathcal{T}_r$: reachable target articles of r.

1 **if** $visited(r)$ **then**
2 **return**
3 $visited(r) \leftarrow True$;
4 $\mathcal{C}_r \leftarrow \{c | r \dashrightarrow c \in \mathcal{E}_{\mathcal{RCH}}\}$ // subcategories of r.
5 **foreach** $c \in \mathcal{C}_r$ **do**
6 $ComputeCost(c)$
7 $\mathcal{A}_r \leftarrow \{p' | p' \in \mathcal{A} \wedge r \Rightarrow p'\}$ // attribute articles belong to r.
8 $fanout(r) \leftarrow |\mathcal{A}_r| + |\mathcal{C}_r|$
9 $\mathcal{T}_r \leftarrow (\cup_{p' \in \mathcal{A}_r} \mathcal{T}_{p'}) \bigcup (\cup_{c \in \mathcal{C}_r} \mathcal{T}_c)$ // reachable target articles of r.
10 **foreach** $p \in \mathcal{T}_r$ **do**
11 $pathcnt(\mathcal{F}_r, p) \leftarrow |\{p' | p' \in \mathcal{A}_r, p \in \mathcal{T}_{p'}\}| + \sum_{c \in \mathcal{C}_r} pathcnt(\mathcal{F}_c, p)$
12 $cost_1 \leftarrow \sum_{p' \in \mathcal{A}_r s.t. p \in \mathcal{T}_{p'}} (log_2(fanout(r)) + log_2(fanout(p')))$
13 $cost_2 \leftarrow \sum_{c \in \mathcal{C}_r} (log_2(fanout(r)) + cost(\mathcal{F}_c, p)) \times pathcnt(\mathcal{F}_c, p)$
14 $cost(\mathcal{F}_r, p) \leftarrow \frac{cost_1 + cost_2}{pathcnt(\mathcal{F}_r, p)}$
15 $cost(\mathcal{F}_r) \leftarrow \sum_{p \in \mathcal{T}_r} cost(\mathcal{F}_r, p) + penalty \times |\mathcal{T} - \mathcal{T}_r|$
16 **return**

The algorithm of searching for k-facet interface is shown in Algorithm 5. To reduce the search space, our algorithm only considers $\mathcal{I}_n$, the top n facets returned from Algorithm 3. We further reduce the space by excluding from consideration those top ranked facets that are subsumed by other top facets (line 1). In other words, we only keep $\mathcal{I}_{n^-}$, the maximal *antichain* of $\mathcal{I}_n$ based on the graph (category hierarchy) subsumption relationship. This is in line with the idea of avoiding large overlap between facets (Section 4.2).

Given $\mathcal{I}_{n^-}$, instead of exhaustively considering all possible k-element subsets of $\mathcal{I}_{n^-}$, we apply a *hill-climbing method* to search for a local optimum, starting from a random k-facet interface $\mathcal{I}_k$. At every step, we try to find a better neighboring solution, where a k-facet interface $\mathcal{I}_{new}$ is a neighbor of $\mathcal{I}_k$ if they only differ by one facet (line 9). Given the $k \times (n-k)$ possible neighbors at every step, we examine the neighbors in the order of their average navigational cost (line 5, 6, and 9). The algorithm jumps to the first encountered better neighbor among the $k \times (n-k)$ neighbors. The algorithm stops when no better neighbor can be found. As the goal function to be optimized in hill-climbing, $\mathcal{I}_{new}$ is considered better if the facets of $\mathcal{I}_{new}$ have both smaller pair-wise similarities and smaller navigational costs than that of $\mathcal{I}_k$ (line 14). The idea of combining similarity and cost is motivated in Section 4.2.

Besides the above hill-climbing method that considers both similarity and cost, we also explore several other heuristics including directly using the top k facets from Algorithm 3, and a hill-climbing method based on similarity only. We also consider a greedy method that adds k facets one by one into the interface. At each step, the facet that results in the smallest average pair-wise similarity is included into the interface. We empirically compare these several heuristics in Section 6.

Algorithm 5: Facets Selection

Input: $\mathcal{I}_n$: the top n $\mathcal{RCH}$-induced facets with the smallest costs.

Output: $\mathcal{I}_k$: a discovered faceted interface with k facets (k<n)

// remove subsumed facets from $\mathcal{I}_n$

1 $\mathcal{I}_{n^-} \leftarrow \{\mathcal{F}_c | \nexists \mathcal{F}_{c'} \in \mathcal{I}_n$ s.t.$\mathcal{F}_c$ is subsumed by $\mathcal{F}_{c'}$, i.e., c is a descendant category of $c'\}$

// hill climbing

2 $\mathcal{I}_k \leftarrow$ a random k-facet subset of $\mathcal{I}_{n^-}$; $\mathcal{I}' \leftarrow \mathcal{I}_{n^-} \backslash \mathcal{I}_k$

3 **repeat**

5 make $\mathcal{I}_k$=<$\mathcal{I}_k[1]$,...,$\mathcal{I}_k[k]$> sorted in increasing order of cost.

6 make $\mathcal{I}'$=<$\mathcal{I}'[1]$,...,$\mathcal{I}'[n-k]$> sorted in increasing order of cost

7 **for** $i = k$ **to** 1 **step** -1 **do**

8 **for** $j = 1$ **to** $n-k$ **do**

9 $\mathcal{I}_{new} \leftarrow (\mathcal{I}_k \backslash \{\mathcal{I}_k[i]\}) \cup \{\mathcal{I}'[j]\}$

10 $S_1 \leftarrow \sum_{\mathcal{F}_c, \mathcal{F}_{c'} \in \mathcal{I}_{new}, \mathcal{F}_c \neq \mathcal{F}_{c'}} sim(\mathcal{F}_c, \mathcal{F}_{c'})$

11 $C_1 \leftarrow \sum_{\mathcal{F}_c \in \mathcal{I}_{new}} cost(\mathcal{F}_c)$

12 $S_2 \leftarrow \sum_{\mathcal{F}_c, \mathcal{F}_{c'} \in \mathcal{I}_k, \mathcal{F}_c \neq \mathcal{F}_{c'}} sim(\mathcal{F}_c, \mathcal{F}_{c'})$

13 $C_2 \leftarrow \sum_{\mathcal{F}_c \in \mathcal{I}_k} cost(\mathcal{F}_c)$

14 **if** *(*$S_1 \leq S_2$ **and** $C_1 < C_2$*)* **or** *(*$S_1 < S_2$ **and** $C_1 \leq C_2$*)* **then**

15 $\mathcal{I}_k \leftarrow \mathcal{I}_{new}$; $\mathcal{I}' \leftarrow \mathcal{I}_{n^-} \backslash \mathcal{I}_k$

16 go to line 5

17 **until** $\mathcal{I}_k$ *does not change* ;

18 **return** $\mathcal{I}_k$

6. EXPERIMENTAL EVALUATION

In this section we describe the details of our experiment settings and present the results.

6.1 Experiment Settings

`Facetedpedia` is implemented in C++ and the dataset is stored in a `MySQL` database. The experiments are executed on Dell PowerEdge 2900 III server running Linux kernel 2.6.27, with dual quadcore Xeon 2.0GHz processors, 2x6MB cache, 8GB RAM, and three 1TB SATA hard drivers in RAID5.

Dataset: We downloaded the `Wikipedia` dump of July 24, 2008 [10] as a `MySQL` database. In particular, we used the tables *page.sql*, *pagelinks. sql*, *categorylinks.sql*, and *redirect.sql*, which provide all the relevant data including the hyperlinks between articles, categories of articles, and the category system. We performed several preprocessing tasks on the tables, including the detection and removal of cycles in the category hierarchy. Although cycles should usually be avoided as suggested by `Wikipedia`, the category hierarchy in the dataset indeed contains a very small number (594) of elementary cycles (intuitively "smallest" cycles) due to various reasons. We applied a modified depth-first search algorithm to detect the elementary cycles. The category hierarchy is made acyclic by removing the last encountered edge in each elementary cycle during the depth-first search. Other performed preprocessing steps include: removing tuples irrelevant to articles and categories; replacing redirect articles by their original articles; removing special articles such as lists and stubs. We also applied performance tuning of the database, including creating additional indexes on *page_id* in various tables.

Queries: We experimented with 20 keyword queries: 1) "Nobel laureate", 2) "action movie", 3) "country singer", 4) "European physicist", 5) "American president", 6) "mountain ranges of Asia", 7) "Turing Award winner", 8) "Ivy League school", 9) "philosopher", 10) "American civil war", 11) "Internet", 12) "economic depression", 13) "database research", 14) "national parks in United States", 15) "retirement plan in United States", 16) "Chinese cuisine", 17) "passenger car", 18) "villains Batman", 19) "superhero", 20) "global warming".

Snippet of a faceted interface generated by `Facetedpedia` for query "country singer" is shown in Figure 17.

Parameters in algorithms: Each of the above 20 queries was sent to `Google` with domain constraint *site:en.wikipedia.org* to get the top 200 (s=200) English `Wikipedia` target articles. The relevant category hierarchy $\mathcal{RCH}$ was then generated by applying Algorithm 2 on the aforementioned `MySQL` database. By default, Algorithm 3 returns top 200 (n=200) facets and Algorithm 5 generates top 10 facets (k=10). The value of $penalty$ in Definition 10 was set as 7. It was empirically selected by investigating the relationship between the number of unreachable target articles ($|\mathcal{T} - \mathcal{T}_r|$) and the total navigational costs of reachable targets ($\sum_{p \in \mathcal{T}_r} cost(\mathcal{F}_r, p)$).

6.2 User Studies

We conducted user studies to evaluate the effectiveness of our facets ranking metrics and interface search algorithms, and to compare the quality of the facets generated by `Facetedpedia` with `Castanet` [21]. Note that the `Castanet` system is used for generating facets from a static collection of documents in a specific domain (e.g., recipes). We applied `Castanet` algorithm on the dynamic set of keyword search result articles. Although not originally designed for such purposes, `Castanet` still appears to be possibly the closest related work.

All user studies were conducted in `Amazon Mechanical Turk` [11]. Users `Mechanical Turk` are required to work on tasks, known as HITs (Human Intelligence Tasks). We designed three different kinds of HITs: (1) studying the quality of a single facet, (2) the quality of a k-facet interface, and (3) the effectiveness of facets ranking metrics. In all three cases, each HIT contains 1 query and 20 different users are required to work on it.

We encountered several challenges in using `Mechanical Turk` for user study. First it does not support interactive tasks such as exploring a dynamically generated interface. Second a complete faceted interface together with the target articles is too large to be placed as a task. Even if a task only requires the user to examine a portion of the interface, a user may not like to take the task as she can feel overwhelmed and might fail to judge them correctly. Users mostly prefer small and quick tasks that can give them credits quickly.

Therefore we adhere to a logical but somewhat more simplistic approach in this user study: in each HIT a user is shown one navigational path in each facet and is asked to evaluate the quality with respect to that given path. More specifically, given a query and a target article, an user is asked to assign a quantitative rating (1 (Not Relevant), (2) Relevant, or (3) Very Relevant) which depicts the effectiveness of her navigation process to that target article following that path, from facet root leading to the target article.

Therefore in (1) single-facet study, a HIT consists of a path generated by `Facetedpedia` and a path generated by `Castanet`; in (2) k-facet interface study, a HIT consists of k=3 paths, each from a different facet, generated by `Facetedpedia` and similarly k=3 paths from `Castanet`. In addition to individually rate these paths, the user also assigns an absolute preference of one system over the other system. In (3) study of effectiveness of ranking metrics, the user compares the results of various `Facetedpedia` algorithms.

(1) Single-Facet Study: The results are shown in Figure 7 and 8. As can be seen from the figures, `Facetedpedia` is unanimously preferred over `Castanet` and has been evaluated to be "Very Relevant" most of the time.

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

[11] www.mturk.com

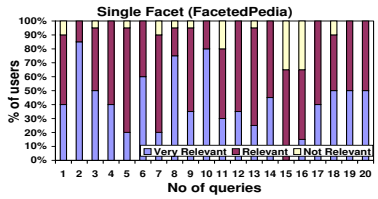


Figure 7: Single-Facet Study of `Facetedpedia`

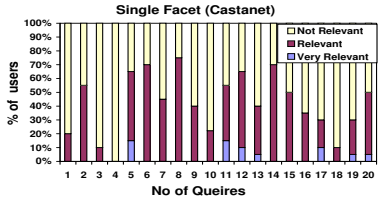


Figure 8: Single-Facet Study of `Castanet`

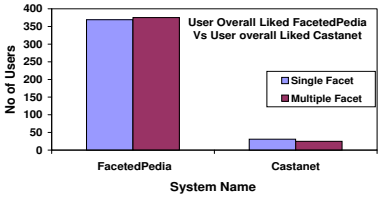


Figure 9: User preference of the Systems

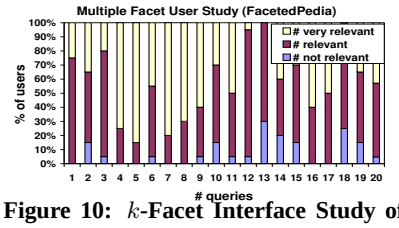


Figure 10: k-Facet Interface Study of `Facetedpedia`

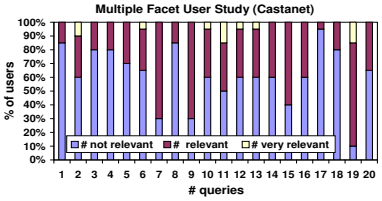


Figure 11: k-Facet Interface Study of `Castanet`

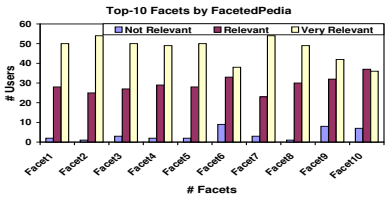


Figure 12: Top-10 Returned Facets

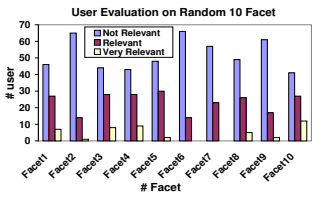


Figure 13: Random-10 Facets

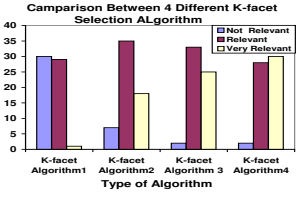


Figure 14: Different K-Facet Selection Algorithms

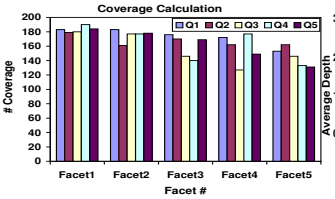


Figure 15: Coverage of Facets Generated by `Facetedpedia`

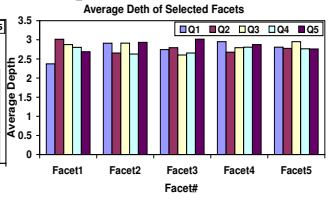


Figure 16: Average Depth of Facets by `Facetedpedia`

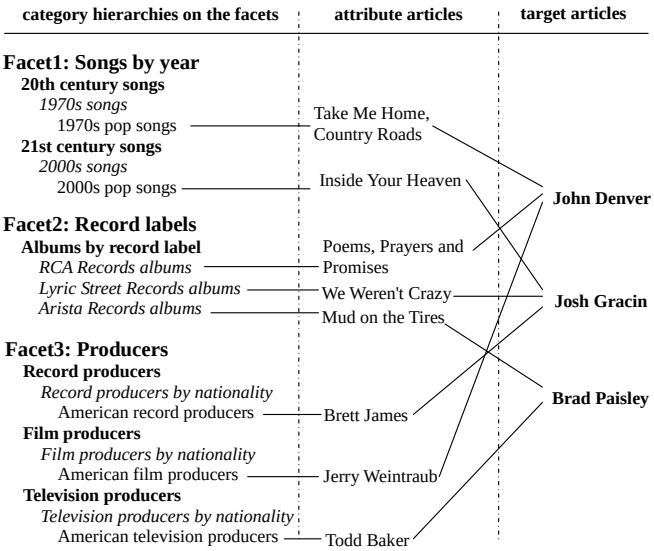


Figure 17: Snippet of a Sample Faceted Generated by `Facetedpedia`

(2) k-Facet Interface Study: The results are shown in Figure 10 and 11. They show that `Facetedpedia` outperforms `Castanet` in generating faceted interfaces. We also present the statistics of overall preference of users between the two systems, combining single facet and k-facet interface studies, in Figure 9. It shows that `Facetedpedia` is unanimously preferred over `Castanet`.

(3) Effectiveness of Ranking Metrics: We compare the quality of top-10 facets generated by `Facetedpedia` and another set of 10 randomly chosen facets. A user rates each set individually. The results from all HITs are aggregated and shown in Figure 12 and Figure 13. The quality of the set from `Facetdpedia` are rated better than the set from Random-10 facets.

We also design HITs for evaluating the quality of k-facet interfaces generated by the 4 different algorithms in Section 5, which are (1) direct top-10, (2) greedy method, (3) hill climbing by similarity only, and (4) hill climbing by both similarity and cost. Figure 14 shows the aggregated results. As can be seen, (1) comes out to be the least effective and (4) gets the best evaluation.

6.3 Depth and Coverage of Generated Facets

In addition to user studies, we also perform internal evaluations to evaluate the quality of the generated facets by `Facetedpedia`. Figure 16 shows the average depth of generated facets, and Figure 15 shows the coverage (the number of reachable target articles) of generated facets by `Facetedpedia`. Figure 16 corroborates the fact that generated facets are shallow thus cheap in navigation. Figure 15 validates that the facets generated by `Facetedpedia` have good coverage in general.

7. 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.

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

[17] 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.

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

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

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

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

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

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

[24] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *WWW*, 2006.

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

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

[27] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. In *WWW*, 1999.

[28] 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.