Facetedpedia: Dynamic Generation of Query-Dependent Faceted Interfaces for Wikipedia

Chengkai Li, Ning Yan, Senjuti B. Roy, Lekhendro Lisham, Gautam Das
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019
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 in Wikipedia. Given the set of Wikipedia articles resulting from a keyword query, Facetedpedia generates a faceted interface for navigating the result articles. Compared with other faceted retrieval systems, Facetedpedia is fully automatic and dynamic in both facet generation and hierarchy construction, and the facets are based on the rich semantic information from Wikipedia. The essence of our approach is to build upon the collaborative vocabulary in Wikipedia, more specifically the intensive internal structures (hyperlinks) and folksonomy (category system). Given the sheer size and complexity of this corpus, the space of possible choices of faceted interfaces is prohibitively large. We propose metrics for ranking individual facet hierarchies by user's navigational cost, and metrics for ranking interfaces (each with k facets) by both their average pairwise similarities and average navigational costs. We thus develop faceted interface discovery algorithms that optimize the ranking metrics. Our experimental evaluation and user study verify the effectiveness of the system.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

faceted search, data exploration, Wikipedia

1. INTRODUCTION

Wikipedia has become the largest encyclopedia ever created, with close to 3 million English articles by far. The prevalent manner in which the Web users access Wikipedia articles is keyword search. Keyword search has been effective in finding specific Web pages matching the keywords. Therefore it may well satisfy the users when they are causally interested in a single topic and use Wikipedia as a dictionary or encyclopedia for that topic. However, Wikipedia has now become a primary knowledge source for many casual users and even an integral component in the knowledge management systems of businesses for decision making. It is thus typical for a user

to explore a set of relevant topics, instead of targeting a particular topic, for more sophisticated information discovery and exploratory tasks. With only keyword search, one would have to digest the potentially long list of search result articles, follow hyperlinks to connected articles, adjust the query and perform multiple searches, and synthesize information manually. This procedure is often time-consuming and error-prone.

One useful mechanism for information exploration is the *faceted interface*, or the so-called *hierarchical faceted categories* (HFC) [9]. A faceted interface for 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 user can navigate an individual facet through its hierarchy of categories and ultimately a specific "property" value if necessary, thus reaching those objects associated with the categories and the value on that facet. The user navigates multiple facets and the intersection of the chosen objects on individual facets are brought to the user's attention. The procedure hence resembles repeated constructions of conjunctive queries with selection conditions on multiple dimensions.

In this paper we propose Facetedpedia[1], a faceted retrieval system which is capable of dynamically generating query-dependent facets for a set of Wikipedia articles. We use the following example to further illustrate.

Example 1 (Motivating Example): Imagine that a user is exploring information about action films. The Facetedpedida system takes a keyword query, say, "us action film", as the input and obtains a ranked list of search result articles. It will create a faceted interface, as shown in Figure 1. The system dynamically derives k facets (region (A)) for covering the top s result articles. For instance, for "us action film", these dimensions (facets) can include Companies, Actors, and so on. Each facet is associated with a hierarchy of categories. Each article can be assigned to the nodes in these hierarchies, with each assignment representing an attribute value of the article. On each facet, the user can navigate through the category path which is formed by parent-child relationships of Wikipedia categories. [2] The interface also shows the navigation paths (region (B)) and article titles (region (C)). When the user clicks one article title, the corresponding Wikipedia article would be shown. (This part of the interface is omitted.)

Here in Figure 1 we only show three facets from the generated interface in region (A): (1) *Film_production_companies_of_the_United _States*; (2) *American_film_actors*; (3) *American_television_actors*. When the user selects any facet items for navigation in region (A), a user navigational path is added in region (B). Here we show only

---

[1] http://idir.uta.edu/facetedpedia/

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

Figure 1: The faceted retrieval interface of Facetedpedia.

one path: *Films_by_subgenre>Action_films_by_genre>Science_fiction_action_films*, which means the user selected facet root *Films_by_subgenre*, then its subcategory *Action_films_by_genre*, and the subcategory of subcategory, *Science_fiction_action_films*. There are thirteen articles satisfying the chosen navigational paths, and they are shown in region (C). In this way, the user filters the large number of result articles and finds those matching her interests. ∎

1.1 Overview of Challenges and Solutions

We study the problem of dynamic discovery of query-dependent faceted interfaces. Given the set of top-s ranked Wikipedia articles as the result of a keyword search query, Facetedpedia produces an interface of multiple facets for exploring the result articles.

We focus on *automatic* and *dynamic* faceted interfaces. The facets could not be pre-computed due to the query-dependent nature of the system. 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 query-independent static faceted interface (either manually or automatically generated) may suffice. By contrast, the articles in Wikipedia are lacking such predetermined dimensions that could fit all possible dynamic query results. Therefore efforts on static facets would be futile. Even if the facets can be pre-computed for some popular queries, say, based on query logs, the computation must be automatic and dynamic. Given the sheer size and complexity of Wikipedia and its rapid growth, a manual approach would be prohibitively time-consuming and cannot scale to stay up-to-date. The main challenges in realizing Facetedpedia are summarized as follows:

Challenge 1: The facets and their category hierarchies are not readily available.

The concept of faceted interface is built upon two pillars: facets (i.e., dimensions or attributes) and the category hierarchy associated with each facet. The definition of "facet" itself for Wikipedia does not arise automatically, leaving alone the discovery of a faceted interface. Therefore we must answer two questions: (1) *facet identification*– What are the facets of a Wikipedia article?; and (2) *hierarchy construction*– Where does the category hierarchy of a facet come from?

Challenge 2: We need metrics for measuring the "goodness" of facets both individually and collectively.

We need to find facets useful for user navigation. A goodness metric for ranking the facets is needed. The problem gets even more complex because the utilities of multiple facets do not necessarily build up linearly– Since the facets in an interface should ideally describe diverse aspects of the result articles, a set of individually "good" facets may not be "good" collectively.

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

It is infeasible to directly apply the ranking metric exhaustively on all possible choices, due to the large search space. Furthermore, the interactions between the facets in a faceted interface make the computation of its exact cost intractable. Even computing the costs of individual facets without considering the interactions is nontrivial, given the size and the complexity of Wikipedia.

1.2 Summary of Contributions and Outline

- **Concept: Faceted Wikipedia.** We propose an automatic and dynamic faceted retrieval system for 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)

Figure 2(a): semantics of hierarchies (rich / shallow); types of dimensions (structured / text)

Endeca Mercado [3,7,17] | Facetedpedia [16]
[8] | [5,6]
[15]*

Figure 2(b): hierarchy construction (automatically discovered / predefined); facet identification (predefined / automatically discovered)

[5] [6] | Facetedpedia [16]
[8]
[3,17] Endeca Mercado | [7] [15]*
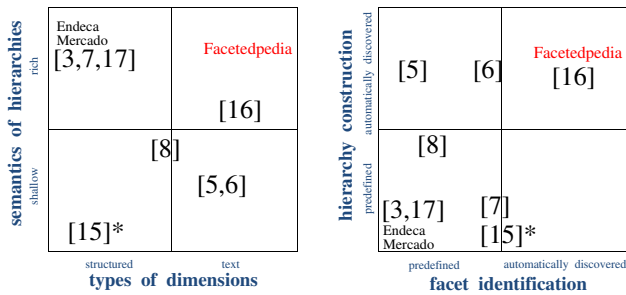
(a) Facet types and semantics. (b) Automation and dynamism.

* The work does not support hierarchy on facets.

Figure 2: Taxonomies of faceted retrieval systems.

- **Metrics: Facet Ranking**. Based on a user navigation model, we propose 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 in the large search space. (Section 5)
- **System Evaluation: Facetedpedia**. We conducted user study to evaluate the effectiveness of the system and to compare with alternative approaches. We also measured its quality and efficiency quantitatively. (Section 6)

2. FACETED RETRIEVAL SYSTEMS: A COMPARATIVE STUDY

Faceted interface has become influential over the last few years and we have seen an explosive growth of interests in its application [11, 17, 9, 17, 9, 16, 6, 5, 14, 15, 8, 7, 3]. Commercial faceted search systems have been adopted by vendors (such as Endeca, IBM, and Mercado), as well as E-commerce Websites (e.g., eBay.com, Amazon.com). The utility of faceted interfaces was investigated in various studies [11, 9, 12, 17, 10, 12, 13, 9], where it was shown that users engaged in exploratory tasks often prefer such result grouping over simple ranked result list (commonly provided by search engines), as well as over alternative ways of organizing retrieval results, such as clustering [4, 18, 10].

In this section we present taxonomies to characterize the relevant faceted retrieval systems and compare them with Facetedpedia. Existing research prototypes or commercial faceted retrieval systems mostly cannot be applied to meet our goals, because they either are based on manual or static facet construction, or are for structured records or text collections with prescribed metadata. Very few have investigated the problem of dynamic discovery of both facet dimensions and their associated category hierarchies.

To the best of our knowledge, we are the first to propose a query-dependent faceted retrieval system for Wikipedia. CompleteSearch [2] supports query completions and query refinement in Wikipedia by a special type of "facets" on three dimensions that are very different from our notion of general facets: query completions matching the query terms; category names matching the query terms; and categories of result articles. Recently, a faceted Wikipedia search interface came out of the DBPedia [1] project around the same time as our work. The facets there appear to be query-independently extracted from common Wikipedia infobox attributes, although the underlying method remains to be proprietary at this moment.

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

Previous systems roughly belong to two groups on this aspect. In some systems the facets are on relational data (e.g., Endeca, Mercado, [15]) or structured attributes in schemata (e.g., [17, 7, 3]) and the hierarchies on attribute values are predefined based on domain-specific taxonomies. The hierarchies could even be manually created, thus could contain rich semantic information. In some other systems a facet is a group of textual terms, over which the hierarchy is built upon thesaurus-based IS-A relationships (e.g., [16]) or frequency-based subsumption relationships between general and specific terms (e.g., [6, 5]). These systems cannot leverage as much semantic information. The work [8] is in the middle of Figure 2(a) since it has both structured dimensions and a subsumption-based topic taxonomy.

In contrast, Facetedpedia enables semantic-rich facet hierarchies (distilled from Wikipedia category system) over text attributes (hyperlinked Wikipedia article titles). In the absence of predefined schemata, it builds facet hierarchies with abundant semantic information from the collaborative vocabulary, instead of relying on IS-A or subsumption relationships.

Figure 2(b): Taxonomy by Degree of Automation and Dynamism

When building the two pillars in a faceted interface, namely the facet and the hierarchy, Facetedpedia is both automatic and dynamic, as motivated in Section 1.1. On this aspect, none of the existing systems could be effectively applied in place of Facetedpedia, because none is fully automatic in both facet identification and hierarchy construction.

In some systems (e.g., Endeca, Mercado, [15, 3, 17, 7]) the dimensions and hierarchies are predefined, therefore they do not discover the facets or construct the hierarchy. In [7, 15] a subset of interesting/important facets are automatically selected from the predefined ones. In [6, 5] the set of facets are predefined, but the hierarchies are automatically created based on subsumption. In [8] only one special 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 [16]. The algorithm is intended for short textual descriptions with limited vocabularies in a specific domain. It automatically creates facets from 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.

3. FACETED INTERFACE FOR WIKIPEDIA BY COLLABORATIVE VOCABULARY

In discovering faceted interfaces for Wikipedia, the basis of our approach is to exploit its user-generated *collaborative vocabulary* such as the "grassroots" category system. Even internal Wikipedia hyperlinks are an instance of collaborative vocabulary in a broader sense, as they indicate the users' collaborative endorsement of relationships between entities. The collaborative vocabulary represents the collective intelligence of many users and rich semantic information, and thus constitutes the promising basis for faceted interfaces. With regard to **the concept of facet dimension**, the Wikipedia articles hyperlinked from a search result article are exploited as its attributes. 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 given article. This view largely enriches the semantic information associated with the result articles. With regard to **the concept of category hierarchy**, the Wikipedia category system provides the category-subcategory relationships between categories, allowing users to go from general to specific when specifying conditions. We now formally define the concepts in our framework and deliver the problem specification.

Definition 1 (Target Article, Attribute Article): Given a keyword query q, the set of top-s ranked Wikipedia articles, $\mathcal{T} = \{p_1, ..., p_s\}$,
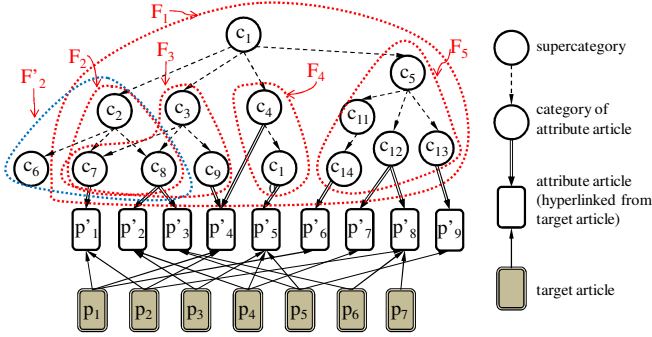
Figure 3: The concept of facet.

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 $p' \leftarrow p$. 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 $p_j \in \mathcal{T}$. ∎

Definition 2 (Category Hierarchy): 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(c)-subcategory(c') relationships. The root category of $\mathcal{H}$, $r_{\mathcal{H}}$, is Category:Fundamental. [3] ∎

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): In Figure 3 there are 7 target articles $(p_1, ..., p_7)$ and 9 attribute articles $(p'_1, ..., p'_9)$. The category hierarchy has 14 categories $(c_1, ..., c_{14})$. The figure highlights 6 facets $(\mathcal{F}_1, ..., \mathcal{F}_5, \text{ and } \mathcal{F}'_2)$. 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 $c_{11} \dashrightarrow c_{14}$ and $c_7 \Rightarrow p'_1$ are unnecessary since there is only one choice under c_{11} 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 the target articles that have the attribute. These target articles are called *reachable target articles*. A *facet* is a *safe reaching facet* 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 of $\mathcal{F}$, starting from c, that reaches p. In order to capture the notion of "reach", we formally define *navigational path* as follows.

Definition 4 (Navigational Path): With respect to the target articles $\mathcal{T}$, the attribute articles $\mathcal{A}$, and a facet $\mathcal{F}(r, \mathcal{C}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$, a *navigational path* in $\mathcal{F}$ is a sequence $c_1 \dashrightarrow ... \dashrightarrow c_t \Rightarrow p' \leftarrow p$, where,

- 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 $\mathcal{F}$.
- $p' \in \mathcal{A}$, 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 $p \to p'$).

[3] http://en.wikipedia.org/wiki/Category:Fundamental

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

Definition 5 (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 3 (Navigational Path and Faceted Interface): Continue the running example. In Figure 3, $I = \{\mathcal{F}_2, \mathcal{F}_5\}$ is a 2-facet interface. Two examples of navigational paths are $c_2 \dashrightarrow c_8 \Rightarrow p'_3 \leftarrow p_5$ and $c_5 \dashrightarrow c_{13} \Rightarrow p'_9 \leftarrow p_5$. 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. FACET RANKING

The search space of the faceted interface discovery problem is prohibitively large. Given the set of s target Wikipedia articles to a keyword query, $\mathcal{T}$, 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 3 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. By definition, any rooted and connected subgraph of $\mathcal{H}$ that safely reaches $\mathcal{T}$ is a candidate facet, 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. [4] 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 4 (Navigation in Faceted Interface): Continue the running example in Figure 3. Consider a faceted interface $I = \{\mathcal{F}_2, \mathcal{F}_5\}$. A sequence of navigational steps on this interface are in Figure 4. At the beginning, the user has not selected any facet to explore, therefore all 7 target articles are available (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
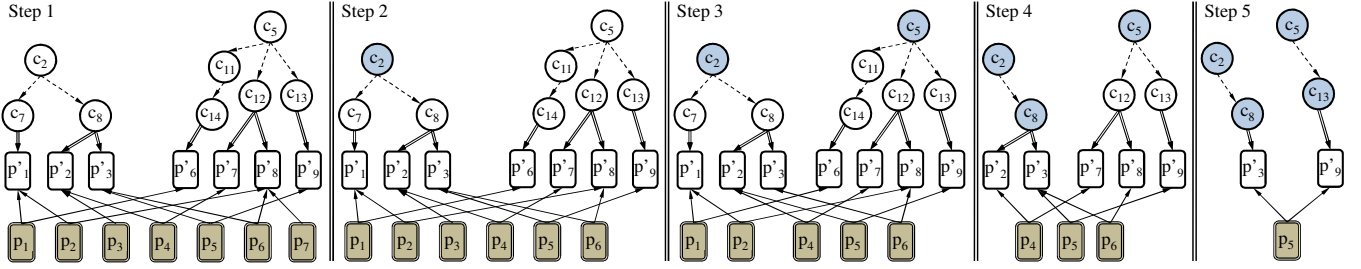
Figure 4: The navigation on a 2-facet interface $\mathcal{I} = \{\mathcal{F}_2, \mathcal{F}_5\}$.

further removes p_3 from consideration (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. Intuitively a low-cost path, i.e., a path that demands small user effort, should have a small number of steps and at each step only require 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., the number of choices) at every step, in logarithmic form. [5]

Definition 6 (Cost of Navigational Path): With respect to the target articles $\mathcal{T}$, the corresponding attribute articles $\mathcal{A}$, and a facet $\mathcal{F}(r, \mathcal{C}_\mathcal{F}, \mathcal{E}_\mathcal{F})$, the cost of a navigational path in $\mathcal{F}$ is

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

where $l = c_1\dashrightarrow ...\dashrightarrow c_t \Rightarrow p' \leftarrow p$.

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

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

where $\mathcal{A}_c$ is the set of attribute articles belonging to 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 $\mathcal{F}$,

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

Note that we made several assumptions for simplicity of the model. The cost formula only captures the "browsing" cost. A full-fledged formula would need to incorporate other costs, such as the "clicking" cost in selecting a choice and the cost of "backward" navigation when the user decides to change a previous selection. Furthermore, we assume the user always completes the navigational path till reaching the target articles. In reality, however, the user may stop in the middle when she already finds desirable articles reachable from the current selection of category. We leave the investigation of more sophisticated models to future study.

---

[5]The intuition behind the logarithmic form is: When presented with a number of choices, the user does not necessarily scan through the choices linearly but by a binary search.

Example 5 (Cost of Navigational Path): We continue the running example. Given $l = c_5\dashrightarrow c_{12} \Rightarrow p_8' \leftarrow p_6$, a navigational path of $\mathcal{F}_5$ in Figure 3, $cost(l) = fanout(c_5) + fanout(c_{12}) + fanout(p_8') = log_2(3) + log_2(2) + log_2(3) = 4.17$. ∎

Albeit the basis of our facet ranking metrics, the definition of navigational cost is not sufficient in measuring the goodness of a facet. It does not consider such a scenario that a facet cannot fully reach all the target articles, which presents an unsatisfactory user experience. In fact, low-cost and high-coverage could be two qualities that compete with each other. On the one hand, a low-cost facet could be one that 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. To combine navigational cost with coverage, we penalize a facet by associating a high-cost *pseudo path* with each unreachable article. We then define the cost of a facet as the average cost in reaching each target article.

Definition 7 (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 the navigational paths that start from 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,

$$cost(\mathcal{F}_r, p) = \frac{1}{|l_p|} \times \sum_{l\in l_p} cost(l) \quad (8)$$

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

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

In Formula 7, $penalty$ is the cost of the aforementioned expensive pseudo path that "reaches" the unreachable target articles, i.e., $\mathcal{T} - \mathcal{T}_r$, for penalizing a facet for not reaching them. Its value is empirically selected (Section 6) and is at least larger than the highest cost of any path to a reachable target article.

Example 6 (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 costs of facets are obtained by Formula 7, with $penalty=7$. $cost(\mathcal{F}_2) = \frac{1}{7} \times (\sum_{p\in\{p_1,p_2,p_3,p_4,p_5,p_6\}} cost(\mathcal{F}_2, p) + penalty \times |\mathcal{T} - \mathcal{T}_{\mathcal{F}_2}|) = \frac{1}{7} \times (16 + 7 \times 1) = 3.286$. $\mathcal{F}_2$ 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 the 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. ∎
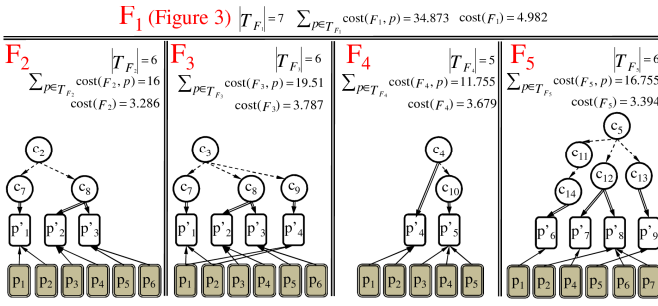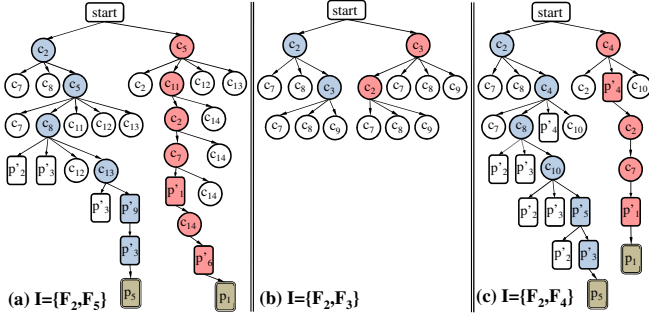
Figure 5: Navigational costs of facets.

(a) I={F₂,F₅} (b) I={F₂,F₃} (c) I={F₂,F₄}

Figure 6: The sequences of navigational steps.

4.2 Multi-Facet Ranking

Even with the cost metrics for individual facets, 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 cheapest k facets. The reason is that when the user navigates multiple facets, the selection made at one facet has impact on the available choices on other facets, as illustrated by Example 4.

To directly follow the approach of ranking faceted interfaces by navigational cost, in principle we could represent the navigational 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 are shown in Figure 6(a). One is c_2, c_5, c_8, c_{13}, p'_9, p'_3, p_5, which are 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 anymore. The other highlighted sequence 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 7 to define the cost of a faceted interface. That entails 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 such exhaustive computation practically infeasible.

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 that overlap much. Imagine a special case when two facets form a subsumption relationship, i.e., the root of one facet is a supercategory of the other root. Presenting both facets would not be desirable since they overlap significantly, thus

cannot capture the expected properties of reaching target articles through different dimensions. As a 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 available choices become $\{c_7, c_8, c_9\}$, which all come from the "dimension", $\mathcal{F}_3$. The same happens if the user selects c_3 and then c_2.

Based on the above observation, we propose to capture the overlap of the k facets by their *average pair-wise similarity*. The pairwise similarity of two facets is the degree of overlap of their category hierarchies and associated attribute articles, defined below.

Definition 8 (Average Similarity of k-Facet Interface): The average pair-wise similarity of a k-facet interface is

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

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 $\mathcal{F}_i$,

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

We choose Jaccard coefficient since it is one of the simplest set-similarity measures. While more complex measures that give different weights to nodes higher in the hierarchy are possible, we do not follow that in the interest of simplicity.

Example 7 (Similarity of Facets): Consider facets $\mathcal{F}_1$, ..., $\mathcal{F}_5$ in Figure 3. $sim(\mathcal{F}_2, \mathcal{F}_3) = \frac{|\mathcal{C}_{\mathcal{F}_2} \bigcap \mathcal{C}_{\mathcal{F}_3}| + |\mathcal{A}_{\mathcal{F}_2} \bigcap \mathcal{A}_{\mathcal{F}_3}|}{|\mathcal{C}_{\mathcal{F}_2} \bigcup \mathcal{C}_{\mathcal{F}_3}| + |\mathcal{A}_{\mathcal{F}_2} \bigcup \mathcal{A}_{\mathcal{F}_3}|}$
$= \frac{|\{c_7, c_8\}| + |\{p'_1, p'_2, p'_3\}|}{|\{c_2, c_7, c_8, c_3, c_9\}| + |\{p'_1, p'_2, p'_3, p'_4\}|} = 5/9$. Other pair-wise similarities can be computed in the same way. The average pari-wise similarity of $\mathcal{I}=\{\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_5\}$ is $sim(\mathcal{I}) = (sim(\mathcal{F}_2, \mathcal{F}_3) + sim(\mathcal{F}_2, \mathcal{F}_5) + sim(\mathcal{F}_3, \mathcal{F}_5))/3 = 5/27$. $\blacksquare$

We do not design a single function to combine the average pair-wise similarity of a faceted interface with its navigational cost, since they represent two measures with different natures. Instead, in Section 5.3 we discuss how to search the space of candidate interfaces by considering both measures.

5. ALGORITHMS

A straightforward approach for 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 challenging optimization problem.

Our k-facet discovery algorithm hinges on (1) reducing the search space; and (2) searching the space effectively and efficiently.

Reducing the Search Space: There are two search spaces in finding a good k-facet interface: 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, $\mathcal{RCH}$-induced facets, which are the facets that contain all the descendant categories of their roots (Section 5.1). To further reduce the space of faceted interfaces, we rank the facets individually by their navigational costs (Section 5.2) and only consider the top ranked facets that do not subsume each other (Section 5.3).

Searching the Space: Instead of exhaustively examining all possible interfaces, we design a hill-climbing based heuristic algorithm to look for a local optimum (Section 5.3). To further tackle the challenge of modeling the interactions of multiple facets in measuring the cost of an interface, the hill climbing algorithm optimizes for both the average navigational cost and the pair-wise similarity of the facets.

Our k-facet discovery algorithm is outlined as three steps: construction of relevant category hierarchy, ranking single facet, and searching for k-facet interface.

5.1 Relevant Category Hierarchy (Algorithm 1)

By Definition 5, the facets in a faceted interface must be safe reaching facets, i.e., they do not contain "dead end" categories that cannot reach any target articles. Therefore the categories appearing in any safe reaching facet could only come from the *relevant category hierarchy* ($\mathcal{RCH}$), which is a subgraph of the Wikipedia category hierarchy $\mathcal{H}$, defined below.

Definition 9 (Relevant Category Hierarchy): Given the category hierarchy $\mathcal{H}(r_\mathcal{H}, \mathcal{C}_\mathcal{H}, \mathcal{E}_\mathcal{H})$, the target articles $\mathcal{T}$, and the 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 $\mathcal{RCH}$. ∎

The procedural algorithm for getting $\mathcal{RCH}$ is in Algorithm 1. Based on definition, straightforwardly we could prove every safe reaching facet of the target articles $\mathcal{T}$ is a (rooted and connected) subgraph of $\mathcal{RCH}$. However, not every rooted and connected subgraph of $\mathcal{RCH}$ is a safe reaching facet. Therefore, even though $\mathcal{RCH}$ is much smaller than $\mathcal{H}$, the search space is still very large which needs us to further shrink the space by considering only one type of safe reaching facets, the $\mathcal{RCH}$-*induced facets*.

Definition 10 ($\mathcal{RCH}$-Induced Facet): Given the relevant category hierarchy $\mathcal{RCH}$ of the 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 8 ($\mathcal{RCH}$ 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 article. $\mathcal{F}_2$ is an $\mathcal{RCH}$-induced facet, but would not be if it does not contain c_7 (or c_8). ∎

Note that every $\mathcal{RCH}$-induced facet is safe reaching, and the single-facet ranking and searching for k-facet are performed on it.

5.2 Ranking Single Facet (Algorithm 2 and 3)

Among all the $\mathcal{RCH}$-induced facets, only the top n facets with the smallest navigational costs are considered in searching for a faceted interface. In ranking the facets by their costs, one straightforward approach is to enumerate all the $\mathcal{RCH}$-induced facets and to separately compute the cost of each facet by enumerating all of its navigational paths. This 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.

To avoid the costly exhaustive method, we design a recursive algorithm that calculates the navigational costs of all the $\mathcal{RCH}$-induced facets by only one pass depth-first search of $\mathcal{RCH}$. The details are in Algorithm 2. The essence of the algorithm is to,

---

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

Algorithm 2: Facet Ranking

Input: $\mathcal{T}$:targets; $\mathcal{A}$:attributes; $\mathcal{RCH}$:relevant category hierarchy.
Output: $\mathcal{I}_n$: top n $\mathcal{RCH}$-induced facets with smallest costs.

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

---

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 such articles by Definition 7. 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 direct subcategories, without accumulating the individual costs of the facets rooted at r's descendants. Therefore it avoids the aforementioned repeated traversal of $\mathcal{RCH}$. More specifically, the lines 11-14 in Algorithm 3 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. We omit the formal correctness proof.

5.3 Searching for k-Facet Interface (Algorithm 4)

Algorithm 4 searches for k-facet interface. To reduce the search space, our algorithm only considers $\mathcal{I}_n$, the top n facets from Algorithm 2. We further reduce the space by excluding those top ranked

Algorithm 3: 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 from $\mathcal{F}_r$; $pathcnt(\mathcal{F}_r, p)$: number of navigational paths reaching p from $\mathcal{F}_r$; $\mathcal{T}_r$: 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.
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**

Algorithm 4: Facet 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 $\mathcal{I}_{n^-}$; $\mathcal{I}' \leftarrow \mathcal{I}_{n^-} \setminus \mathcal{I}_k$
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             $\mathcal{I}_{new} \leftarrow (\mathcal{I}_k \setminus \{\mathcal{I}_k[i]\}) \cup \{\mathcal{I}'[j]\}$
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                 $\mathcal{I}_k \leftarrow \mathcal{I}_{new}$; $\mathcal{I}' \leftarrow \mathcal{I}_{n^-} \setminus \mathcal{I}_k$
16   go to line 5
17 **until** $\mathcal{I}_k$ *does not change* ;
18 **return** $\mathcal{I}_k$

---

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 them in the order of average navigational costs (line 5, 6, and 9). The algorithm jumps to the first encountered better neighbor. 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 considering both similarity and cost is motivated in Section 4.2.

6. EXPERIMENTAL EVALUATION

6.1 Experimental Settings

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

Dataset: We downloaded the Wikipedia dump of July 24, 2008 from http://download.wikimedia.org and loaded the data into 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 system in Wikipedia contains a very small number (594 in the dataset) of elementary cycles [6] due to various reasons. We applied 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 basic performance tuning of the database, including creating additional indexes on *page_id* in various tables. The characteristics of the dataset are summarized in Figure 7. The total size of the tables is 1.2GB.

Queries: We experimented with 20 keyword queries that we designed (Figure 8), in addition to the open queries that the users came up with during user study (Section 6.2).

Parameters in algorithms: Each query was sent to Google with site 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 1 on the aforementioned MySQL database. By default, Algorithm 2 returns top 200 (n=200) facets and Algorithm 4 generates 10 facets (k=10). The value of $penalty$ in Definition 7 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 Facetedpedia, and to compare the quality of the faceted interfaces gener-

---

[6]A cycle is elementary if no vertices in the cycle (except the start/end vertex) appear more than once.

number of articles	$2,445,642$
number of hyperlinks between articles	$109,165,108$
average number of hyperlinks per article	45
number of distinct categories	$329,007$
average number of categories per article	3
number of category-subcategory relationships	$731,097$

Figure 7: Characteristics of the dataset.

$Q1$	action film	$Q2$	country singer
$Q3$	philosophers	$Q4$	Texas universities
$Q5$	Turing Award winner	$Q6$	missile
$Q7$	Ivy League schools	$Q8$	NBA players
$Q9$	historic landmarks	$Q10$	cartoon characters
$Q11$	Microsoft acquired game companies	$Q12$	stand up comedian
$Q13$	graph theorists	$Q14$	lakes in North America
$Q15$	American presidents	$Q16$	battle far east
$Q17$	waterfall national park	$Q18$	Chinese cuisine
$Q19$	premier league clubs	$Q20$	PS3 game

Figure 8: Experiment queries.

ated by Facetedpedia and Castanet [16]. We obtained the implementation of Castanet from its authors. Note that Castanet is intended for static, short, and domain-specific documents with limited vocabularies. Nevertheless, we applied Castanet on the dynamic keyword search results. Although not originally designed for such purposes, Castanet still appears to be possibly the closest related work. We use the same graphical user interface for both systems, to make the comparison irrelevant to interface design.

The user studies were conducted online. The users all have college degrees or are in college, including university students, faculty, staff, and financial and IT company workers. We believe these users are experienced with Web search and comfortable with more sophisticated access mechanisms, matching the target users of our system. To reduce the overhead of the user, we partitioned the 20 queries in Figure 8 into 4 equal-size groups and asked each user to only participate in the 5 queries of one group. For each query group, we sent user-study invitations to roughly equal number of people. Ultimately we were able to collect opinions from totally 36 users, 8 each for 2 groups, and 10 each for the other 2 groups.

For each query, we showed the query keywords and objective description to the user, and asked the user to explore two interfaces pre-generated by Facetedpedia and Castanet, respectively. At the end of each query, the user was asked to provide response to 3 questions, namely $R1$-$R3$ in Figure 9. The available choices for $R1$ and $R2$ are ratings from 1:"useless" to 5:"very useful". The choices for $R3$ are "Facetedpedia" and "Castanet". The same process iterated through the 5 queries in the group assigned to the user. After the 5 queries were done, the user was also provided opportunity to try arbitrary open queries on Facetedpedia, and provided answers to questions $R4$-$R6$ in Figure 9. The available choices are ratings from 1:"strongly disagree" to 5:"strongly agree". The same open query study, however, was not possible for Castanet because the implementation we obtained from the authors takes about 5 minutes to process each query and therefore could not be used for dynamic queries. The reason is that it checks WordNet for each word in constructing category hierarchy. (Remember it was designed for static collection of short texts.)

In Figure 10, column 2 and column 3 records average user ratings per query on questions $R1$ and $R2$ respectively. Column 4 and 5 represent user's absolute preference on one system over the other. Clearly, from the results, Facetedpedia receives much stronger feedback than Castanet on $R1$ and $R2$. Also, for absolute preference, user prefers Facetedpedia over Castanet almost unanimously. Fig-

Choices– 1: useless; 2: not very useful; 3: useful to some extent; 4: useful; 5: very useful	
R1	My rating about usefulness of Facetedpedia.
R2	My rating about usefulness of Castanet.
Choices– Facetedpedia; Castanet	
R3	Which interface is better than the other?
Choices– 1: strongly disagree; 2: disagree; 3: neutral; 4: agree; 5: strongly agree	
R4	The facets in Facetedpedia conveys important concepts regarding the articles related to the query.
R5	Facetedpedia is useful for browsing and exploration purposes.
R6	I look forward to use this interface even in the future for exploratory browsing purposes.

Figure 9: User study questions and available answers.

	Average R1	Average R2	R3-Facetedpedia	R3-Castanet
$Q1$	3.5	2.5	7	1
$Q2$	3.5	2.625	5	3
$Q3$	3.5	2.875	5	3
$Q4$	3.625	2.5	7	1
$Q5$	3.375	2.5	7	1
$Q6$	3.625	3.375	6	2
$Q7$	4.0	3.625	5	3
$Q8$	3.75	3.625	4	4
$Q9$	4.125	3.25	7	1
$Q10$	3.5	3.875	4	4
$Q11$	4.2	3.1	9	1
$Q12$	3.8	3.2	8	2
$Q13$	3.8	3.5	6	4
$Q14$	3.7	3.5	6	4
$Q15$	3.7	3.7	6	4
$Q16$	3.9	2.9	9	1
$Q17$	4.1	3.1	9	1
$Q18$	4.2	2.9	9	1
$Q19$	3.7	2.7	7	3
$Q20$	3.6	3.1	6	4

Figure 10: Usefulness of Facetedpedia and Castanet.

ure 11 records average user ratings per group for $R4$, $R5$ and $R6$. As it can be seen, majority of the groups provide strong positive opinion the about usefulness of facets and the interface generated by Facetedpedia and they believe Facetedpedia interface is effective for exploration purposes.

6.3 Characteristics of Generated Facets

Our experiments compared the effectiveness of three algorithms: *hill-climbing* (Algorithm 4), *top-k*– selecting the top k facets ranked by Algorithm 2, and *random-k*– randomly choosing k facets. Figure 13 shows the average characteristics of the faceted interfaces generated by these methods. Although *hill-climbing* had a slightly worse target article coverage than the other two (5% less), it outperformed them in pair-wise similarity which means the k facets selected have smaller overlap of navigational paths. The detailed tracing results show that *hill-climbing* started from choosing top-k facets and gradually replaced similar facets by less similar ones. The final k facets selected by *hill-climbing* usually were still within the top 30%, while the ones selected by *random-k* were evenly distributed among the results from single-facet ranking. The average width and height of the facets generated by the three methods were about the same, except that *random-k* occasionally chose some much wider facets. Their average width and height were usually around 10 and 6, respectively. Therefore the fanout of internal nodes and the length of navigational paths are within a reasonable range for the users. Overall, *hill-climbing* helps us reducing overlapping facets without losing much coverage of target articles.

6.4 Efficiency Evaluation

We evaluated the scalability of our approach by measuring the average execution time of discovering $k=10$ facets for varying number of target articles (s from 50 to 500). As can be seen from Figure
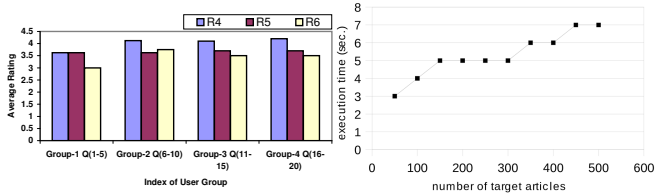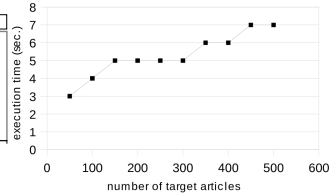
Figure 11: User experience with Facetedpedia for open queries.

Figure 12: Execution time of Facetedpedia vs. number of target articles

	Coverage	average width	average height	average pair-wise similarity
Random-k	72.3%	53.8	8.6	0.108
Top-k	73.9%	10.2	5.5	0.187
Hill-climbing	68.9%	9.8	5.7	0.072

Figure 13: Compare the quality of faceted interfaces generated by various methods.

12, Facetedpedia scales well since the execution time increases linearly by the number of target articles. It also shows that Facetedpedia already achieved fairly fast response without much performance optimization. In average it took 3 seconds to discover the facets for 50 target articles, and 5 seconds for 200 target articles.

7. DISCUSSION

The faceted interfaces generated by Facetedpedia are certainly not perfect and could be improved on many aspects. The pitfalls and drawbacks of our system pose several open challenges which could possibly form new research directions. It is our plan to forward our investigation along the following lines:

First, hyperlinks in Wikipedia articles are not always good features of the target articles. In many cases the hyperlinked articles are important attribute articles that are strongly related to the target articles. However, there are also cases in which the authors of an article make hyperlinks to other articles not because they have strong relationships with the target articles. The author may believe that the readers would not be familiar with an entity mentioned in an article, therefore decides to make that mention an anchor text linking to the article describing the entity. The hyperlinked article is not necessarily highly related to the target article. For example, in Wikipedia article *Independence_Day_(film)*, hyperlinked articles such as *Will_Smith* and *20th_Century_Fox* are certainly valid attribute articles, while *Moon* and *Mexico* may not be. To assure the quality of the discovered facets, we plan to investigate data mining methods and NLP techniques in finding truly related articles.

Second, we found through experiments that a category hierarchy based on both rich semantics and strong IS-A relationships will provide more accurate facets than the current Wikipedia category system. This "grass-roots" folksonomy in Wikipedia, albeit containing user-generated categories with richer semantics than a thesaurus such as WordNet , is not always organized by rigorous IS-A relationships. For instance, it includes subcategories such as *People_from_Texas* and *History_of_Texas* under category *Texas*, which can be misleading to a user who plans to navigate through geographical concepts by choosing *Texas*. We plan to refine the category hierarchy for strong IS-A relationships.

Third, we need to design methods to improve the diversity of the top ranked facets generated by Facetedpedia. Since our ranking metric penalizes the facets that have small coverage, the top ranked ones may tend to come from relatively large concept domains such

as people, organizations, etc. To avoid missing useful facets from small concept domains, one idea is to first cluster the attribute articles into several groups and then make sure that each group has at least a number of facets in the final results.

8. CONCLUSION

In this paper we proposed Facetedpedia, a faceted search system over Wikipedia. This system provides a dynamic and automated faceted search interface for users to browse the articles that are the result of a keyword search query. Given the sheer size and complexity of Wikipedia and the large space of possible faceted interfaces, we proposed metrics for ranking faceted interfaces as well as efficient algorithms for discovering them. Our experimental evaluation and user study verify the effectiveness of our methods in generating useful faceted interfaces. Moreover, our findings pose several open problems for future study. It would also be interesting to further investigate if the proposed framework and methods can be applied to other applications, or even the generic Web.

9. REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] W. Dakka and P. Ipeirotis. Automatic extraction of useful facet hierarchies from text databases. *ICDE*, 2008.

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

[7] D. Debabrata, R. Jun, N. Megiddo, A. Ailamaki, and G. Lohman. Dynamic faceted search for discovery-driven analysis. In *CIKM*, pages 3–12, 2008.

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

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

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

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

[12] 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.

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

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

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

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

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

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