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.basu,roy,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 Facetedpedia 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.
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 path, 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.

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 non-trivial, 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)
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, \( T = \{ p_1, ..., p_s \} \),
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 \( p' \leftarrow p \). Given \( \mathcal{T} \), the set of attribute articles is \( \mathcal{A} = \{ p'_1, \ldots, p'_m \} \), 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 \( \mathcal{H}(r_H, \mathcal{C}_H, \mathcal{E}_H) \), where the node set \( \mathcal{C}_H = \{ c \} \) is the set of categories and the edge set \( \mathcal{E}_H = \{ e_{\leftarrow c} \} \) is the set of category-c-category relationships. The root category of \( \mathcal{H} \), \( r_H \), is Category:Fundamental.

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

**Example 2 (Running Example):** In Figure 3 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 \( \{ F_1, \ldots, F_5, F'_2 \} \). For instance, \( F_2 \) is rooted at \( c_2 \) and consists of 3 categories \( \{ c_2, c_7, c_8 \} \) and 2 edges \( \{ c_2 \leadsto c_7, c_2 \leadsto c_8 \} \). 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 \( c_{11} \leadsto c_{14} \) and \( c_7 \leadsto p'_2 \) 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 \( \forall e \in \mathcal{E}_F \), there exists a target article \( p \in \mathcal{T} \) such that \( e \) reaches \( p \), i.e., there exists \( c \leadsto \ldots \leadsto p' \leftarrow p \), 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 \( \mathcal{F}(r, \mathcal{C}_F, \mathcal{E}_F) \), a navigational path in \( \mathcal{F} \) is a sequence \( c_1 \leadsto \ldots \leadsto c_l \leadsto p' \leftarrow p \), where:

- for \( 1 \leq i \leq l \), \( c_i \in \mathcal{C}_F \), i.e., \( c_i \) is a category in \( \mathcal{F} \);
- for \( 1 \leq i \leq l-1 \), \( c_i \leadsto c_{i+1} \in \mathcal{E}_F \), i.e., \( c_{i+1} \) is a subcategory of \( c_i \) (in category hierarchy \( \mathcal{H} \)) and that category-subcategory relationship is kept in \( \mathcal{F} \);
- \( p' \in \mathcal{T} \), and \( c_l \) is a category of \( p' \) (represented as \( c_l \Rightarrow p' \));
- \( p \in \mathcal{T} \), and \( p' \) is an attribute article of \( p \) (i.e., there is a hyperlink \( p \rightarrow p' \)).

Given a navigational path \( c_1 \leadsto \ldots \leadsto c_l \leadsto p' \leftarrow p \), we say that the corresponding category path \( c_1 \leadsto \ldots \leadsto c_l \) reaches target article \( p \) through attribute article \( p' \), and we also say that category \( c_i \) (for any \( 1 \leq i \leq l \)) reaches \( p \) through \( p' \). Interchangeably we say \( p \) is reachable from \( c_i \) (for any \( 1 \leq i \leq l \)).

**Definition 5 (Faceted Interface):** Given a keyword query \( q \), a faceted interface \( \mathcal{I} = \{ \mathcal{F}_i \} \) is a set of safe reaching facets of the target articles \( \mathcal{T} \). That is, \( \forall \mathcal{F}_i \in \mathcal{I}, \mathcal{F}_i \) safely reaches \( \mathcal{T} \).

**Example 3 (Navigational Path and Faceted Interface):** Continue the running example. In Figure 3, \( \mathcal{I} = \{ \mathcal{F}_3, \mathcal{F}_2 \} \) is a 2 facet-interface. Two examples of navigational paths are \( c_2 \leadsto c_7 \leadsto p'_2 \leftarrow p_5 \) and \( c_6 \leadsto c_{13} \leadsto p'_9 \leftarrow p_6 \). However, \( \{ \mathcal{F}_2, \mathcal{F}_3 \} \) 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 \( \mathcal{H}(r_H, \mathcal{C}_H, \mathcal{E}_H) \), 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. 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 \( \mathcal{I} = \{ \mathcal{F}_2, \mathcal{F}_3 \} \). 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 \( c_2, p_5 \) is filtered out since it is unreachable from \( \mathcal{F}_2 \) (step 2). The user then selects \( c_5 \), which

---

Section 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 navigational cost of a facet as the average cost of its navigational.

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

$$\text{cost}(l) = \log_2(\text{fanout}(p')) + \sum_{c \in \{c_1, \ldots, c_k\}} \log_2(\text{fanout}(c))$$

where $l = c_1 \rightarrow c_2 \rightarrow \cdots \rightarrow c_l \rightarrow p' \leftarrow p$.

In Formula 1, $\text{fanout}(p')$ is the number of (directly) reachable target articles through the attribute article $p'$.

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

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

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

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

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

$$\mathcal{A}_c = \{p | p \in \mathcal{A} \land c \Rightarrow p'\}$$

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

$$\mathcal{C}_c = \{c' : c' \in \mathcal{E}_F \land c \rightarrow c' \in \mathcal{E}_F\}$$

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.

**Example 5 (Cost of Navigational Path):** We continue the running example. Given $l = c_5 \rightarrow c_12 \rightarrow p_6 \leftarrow p_4$, a navigational path of $\mathcal{F}_5$ in Figure 3, $\text{cost}(l) = \text{fanout}(c_5) + \text{fanout}(c_{12}) + \text{fanout}(p_6) = \log_2(3) + \log_2(2) + \log_2(3) = 4.17$.

Although 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 reach all the target articles, which presents an unsatisfactory user experience. In fact, low-cost and high-coverage can 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 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 $\mathcal{F}(r, \mathcal{C}_F, \mathcal{E}_F)$, $\text{cost}(\mathcal{F}_r)$, 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 $\text{penalty}$.

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

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

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

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

In Formula 7, $\text{penalty}$ 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 $\text{penalty} = 7$. $\text{cost}(\mathcal{F}_2) = \frac{1}{6} \times \sum_{p \in \{p_1, p_2, p_3, p_4, p_5, p_6\}} \text{cost}(\mathcal{F}_2, p) + \text{penalty} \times |\mathcal{T} - \mathcal{T}_2| = \frac{1}{6} \times (16 + 7 \times 1) = 3.286$. $\mathcal{F}_2$ 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 $p_7$). $\mathcal{F}_5$ is even more costly due to its wider and deeper hierarchy, although it reaches all target articles.
cannot capture the expected properties of reaching target articles through different dimensions. As a concrete example, consider the navigational steps of $F_2$ and $F_3$ in Figure 6(b). After the user selects $c_2$ from $F_2$ and then $c_3$ from $F_3$, the available choices become \{c_7, c_8, c_9\}, which all come from the “dimension”, $F_3$. 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 pair-wise 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

$$
    \text{sim}(I = \{F_1, \ldots, F_k\}) = \frac{\sum_{1 \leq i < j \leq k} \text{sim}(F_i, F_j)}{k(k-1)/2},
$$

where $\text{sim}(F_i, F_j)$ is defined by the Jaccard coefficient,

$$
    \text{sim}(F_i, F_j) = \frac{|C_{F_i} \cap C_{F_j}| + |A_{F_i} \cap A_{F_j}|}{|C_{F_i} \cup C_{F_j}| + |A_{F_i} \cup A_{F_j}|}
$$

where $C_{F_i}$ is the set of categories in $F_i$ (Definition 3) and $A_{F_i}$ is the set of attribute articles reachable from $F_i$.

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 $F_1, \ldots, F_5$ in Figure 3. $\text{sim}(F_2, F_3) = \frac{|C_{F_2} \cap C_{F_3}| + |A_{F_2} \cap A_{F_3}|}{|C_{F_2} \cup C_{F_3}| + |A_{F_2} \cup A_{F_3}|} = \frac{5}{9}$. Other pair-wise similarities can be computed in the same way. The average pair-wise similarity of $I = \{F_2, F_3, F_5\}$ is $\text{sim}(I) = (\text{sim}(F_2, F_3) + \text{sim}(F_2, F_5) + \text{sim}(F_3, F_5))/3 = 5/27$.

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 $H$ and apply the ranking metrics directly to find the best interface. Such a naive method results in the exhaustive examination of all possible combinations of $k$ instances of all possible facets, i.e., rooted and connected subgraphs of $H$. Clearly it is a prohibitively large search space, given the sheer size and complexity of Wikipedia. The naive 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, 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 (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 \( T \), and the attribute articles \( A \), the relevant category hierarchy (RCH) of \( T \) is a subgraph of \( \mathcal{H} \). Given any category in RCH, it is either directly a category of some attribute article \( p' \in A \) or a supercategory or ancestor of such categories. There exists an edge (category-subcategory relationship) between two categories in RCH if the same edge exists in \( \mathcal{H} \). By this definition the root of \( \mathcal{H} \) is also the root of RCH.

The procedural algorithm for getting RCH is in Algorithm 1. Based on definition, straightforwardly we could prove every safe reaching facet of the target articles \( T \) is a (rooted and connected) subgraph of RCH. However, not every rooted and connected subgraph of RCH is a safe reaching facet. Therefore, even though 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 RCH-induced facets.

Definition 10 (RCH-Induced Facet): Given the relevant category hierarchy RCH of the target articles \( T \), a facet \( F(r, C_F, E_F) \) is RCH-induced if it is a rooted induced subgraph of RCH, i.e., in \( F \) all the descendants of the root \( r \) and their category-subcategory relationships are retained from RCH.

Example 8 (RCH and RCH-Induced Facet): Continue the running example. In Figure 3, the RCH contains all the categories in the category hierarchy \( \mathcal{H} \) except \( c_6 \) (and thus the edge \( c_2 \rightarrow c_6 \)), since \( c_6 \) cannot reach any target article. \( F_2 \) is an RCH-induced facet, but would not be if it does not contain \( c_2 \) (or \( c_6 \)).

Note that every 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 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 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 RCH, because the 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 RCH-induced facets by only one pass depth-first search of RCH. The details are in Algorithm 2. The essence of the algorithm is to, during the recursive traversal of 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 based on the book-kept 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 RCH. More specifically, the lines 11-14 in Algorithm 3 are for computing \( \text{cost}(F, p) \) 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 \( \text{cost}_1 \), the total cost of all the navigational paths \( r \rightarrow p \rightarrow p' \) to \( p \), i.e., the ones that reach \( p \) without going through any other categories; line 13 computes \( \text{cost}_2 \), the total cost of all the navigational paths that go through other categories, by utilizing \( \text{cost}(F, p) \) and \( \text{pathlen}(F, p) \) 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
facets that are subsumed by other top facets (line 1). In other words, we only keep \( I_n \) - the maximal antichain of \( I_n \) 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 \( I_n \), instead of exhaustively considering all possible k-element subsets of \( I_n \), we apply a hill-climbing method to search for a local optimum, starting from a random k-facet interface \( I_k \). At every step, we try to find a better neighboring solution, where a k-facet interface \( I_{new} \) is a neighbor of \( I_k \) 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, \( I_{new} \) is considered better if the facets of \( I_{new} \) have both smaller pair-wise similarities and smaller navigational costs than that of \( I_k \) (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, 3GB RAM, and three 1TB SATA hard drives in RAID5.

Dataset: We downloaded the Wikipedia dump of July 24, 2008 from http://download.wikipedia.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 (504 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 (n=200) English Wikipedia target articles. The relevant category hierarchy \( RCH \) 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 (\(|T - T_r|\)) and the total navigational costs of reachable targets (\( \sum_{p \in T_r} \text{cost}(F_r, p) \)).

6.2 User Studies

We conducted user studies to evaluate the effectiveness of Facetedpedia, and to compare the quality of the faceted interfaces gener-
Which interface is better than the other?

Facetedpedia is useful for browsing and exploration purposes.

I look forward to using this interface even in the future for

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 11, the runtime curve of our approach is sub-quadratic for both Facetedpedia and Castanet. However, our approach is still slower than Castanet for large number of target articles.
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