Dynamic Discovery of Query-Dependent Faceted Interface for Document Exploration

NING YAN, The University of Texas at Arlington
CHENGKAI LI, The University of Texas at Arlington
SENJUTI BASU ROY, The University of Texas at Arlington
GAUTAM DAS, The University of Texas at Arlington

In this paper we investigate methods for dynamically discovering query-dependent faceted interfaces over text documents. Given a set of result documents from a keyword search query, the objective is to produce a faceted interface for exploring the result documents. Different from previous approaches, we aim at developing methods that are fully automatic and dynamic in both facet dimension generation and category hierarchy construction. Toward this goal, we propose a general faceted search model for document exploration. This model is instantiated into two prototype systems, Facetedpedia and Facetednews, for exploring Wikipedia articles and news articles, respectively. Our model utilizes the collaborative vocabularies in Wikipedia, such as its category hierarchy and intensive internal hyperlinks, for building faceted interfaces. Given the sheer size and complexity of Wikipedia data, the search space of possible choices of faceted interfaces is prohibitively large. We proposed metrics for ranking individual facet hierarchies by user navigational cost, and metrics for ranking interfaces (each with k facet hierarchies) by both average pair-wise facet similarities and average navigational costs. We thus developed faceted interface discovery algorithms that optimize for these ranking metrics. Our experimental evaluation and user studies verified the effectiveness of the proposed metrics, the algorithms, and the prototype systems.

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

General Terms: Algorithms, Design, Experimentation, Performance

Additional Key Words and Phrases: faceted search, information exploration, Wikipedia

ACM Reference Format:
Yan, N., Li, C., Roy, S.B., Das, G. 2011. Dynamic Discovery of Query-Dependent Faceted Interface for Document Exploration ACM Trans. Web V, N, Article A (January YYYY), 32 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Faceted search [Hearst 2006] is a useful technique for information exploration, especially when a user needs to browse through a long list of articles or objects, which, without necessary auxiliary facility, could be time consuming and painstaking. 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

This material is based upon work partially supported by NSF Grant IIS-1018865. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of NSF. This article is an enhanced version of the authors' papers in WWW2010 [Li et al. 2010] and CIKM2010 (demo) [Yan et al. 2010].
Author's addresses: Yan, N. Computer Science & Engineering Department, The University of Texas at Arlington; Li, C. Computer Science & Engineering Department, The University of Texas at Arlington; Roy, S.B. Computer Science & Engineering Department, The University of Texas at Arlington; Das, G. Computer Science & Engineering Department, The University of Texas at Arlington.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

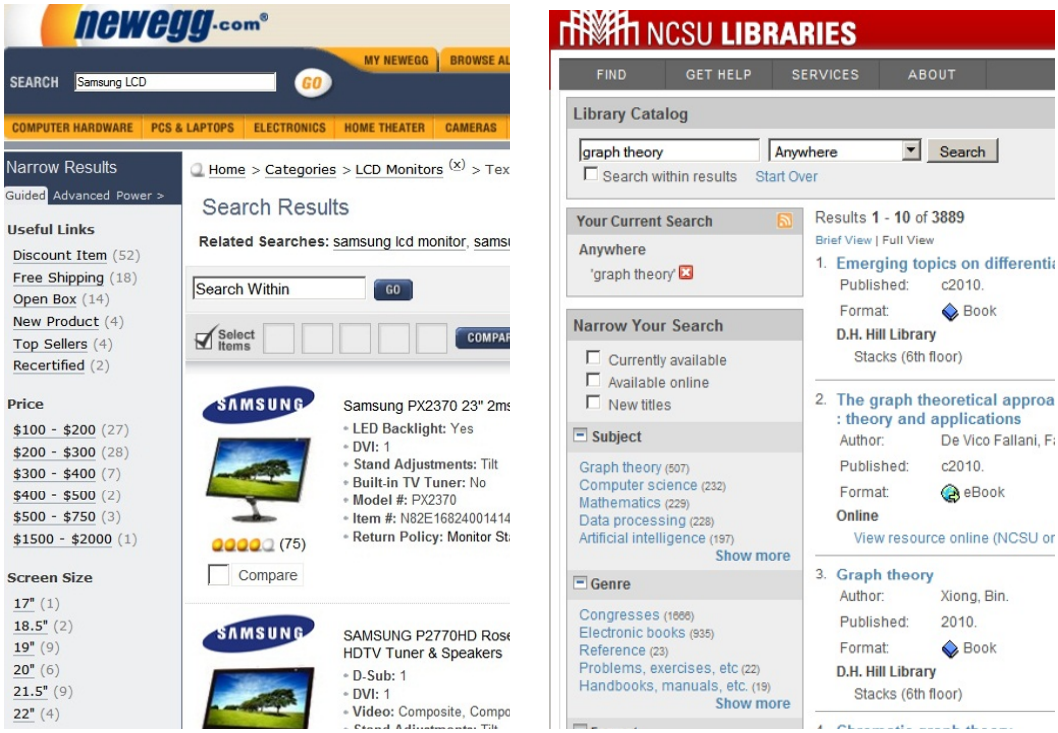


Fig. 1: The faceted search interfaces of newegg.com (left) and NCSU library catalog (right).

categories and ultimately a specific facet value if necessary, thus reaching those objects associated with the chosen categories and 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.

We often experience faceted interfaces when shopping at E-commerce websites such as Amazon.com and Newegg.com. For example, when a customer is shopping for LCDs on Newegg.com, she can first issue a keyword query such as "Samsung LCD". The website will then return a list of LCDs. In addition, a faceted interface will be generated on the resulting web page, for exploring the LCDs (see Figure 1 (left)). The customer can narrow down her search results by facets on dimensions such as *price* and *screen size*. Today, many systems can generate faceted interfaces for relational data such as the product catalogs behind online stores similar to Newegg.com. Similar faceted interfaces are used in searching library catalogs by metadata fields such as *genre*, *subject*, etc. One such example is http://www.lib.ncsu.edu/catalog/ (Figure 1 (right)).

However, many document collections do not come with structured schema or metadata. Hence facets must be discovered from the text content of documents. There are only few such systems that discover faceted interface for document exploration. Furthermore, no prior system dynamically discovers query-dependent interfaces for the resulting documents of keyword search. In this paper we propose a framework for dynamic discovery of query-dependent faceted interface from text

Fig. 2: The faceted search interface of Facetedpedia generated for keywords "*us action film*".

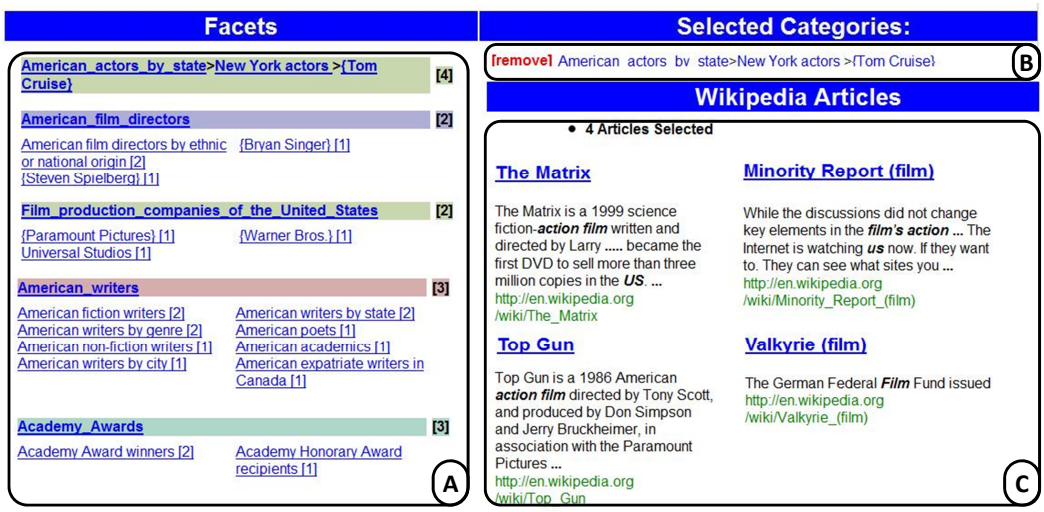
documents. The framework is instantiated into two systems: Facetedpedia [1] [Li et al. 2010; Yan et al. 2010] and Facetednews [2], which are for exploring Wikipedia and news articles, respectively.

1.1. Motivating Examples

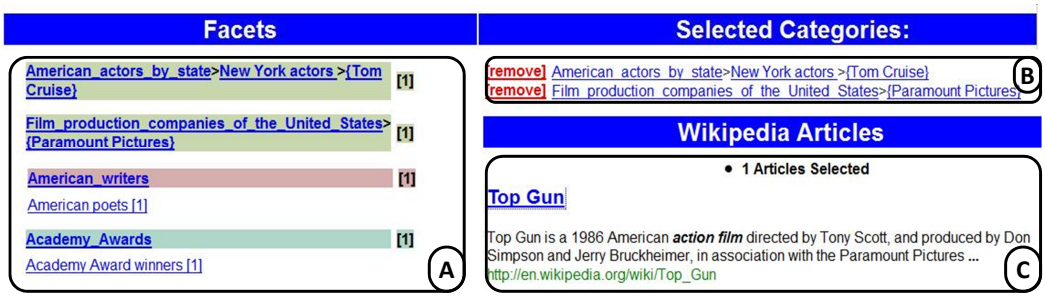
Wikipedia has become the largest encyclopedia ever created, whose pages have grown over 3.5 million English articles. The prevalent manner in which web users access Wikipedia articles is keyword search. Keyword search has been effective in finding specific web pages matching the keywords. Therefore it may well satisfy users when they are casually 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 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, perform multiple searches, and synthesize information manually. This procedure is often time-consuming and error-prone. A faceted interface can facilitate the process of exploring articles in Wikipedia. We use the following example to illustrate.

Example 1 (Motivating Example) *Imagine that a user is exploring information about action films in Wikipedia. The Facetedpedia system takes a keyword query, say, "us action film", as the input and obtains a ranked list of Wikipedia articles from an external search engine. It will create a faceted interface, as shown in Figure 2, for navigating these articles. The system dynamically derives k facets (region (A)) for covering the top s result articles (region(C)) (s=234 in the example). Figure 2 only*

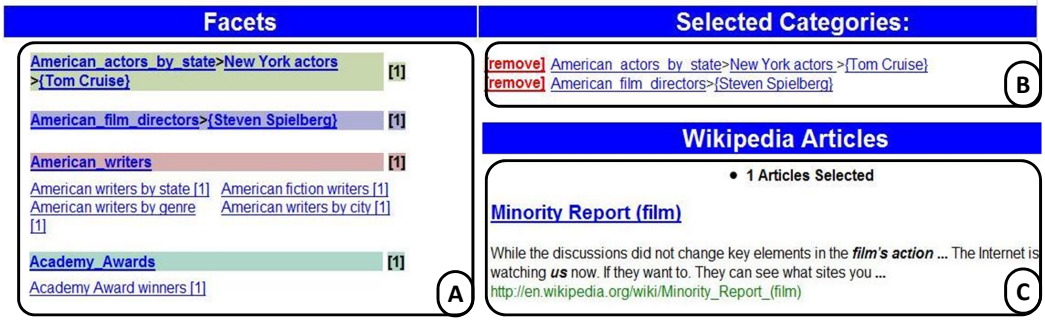
[1] http://idir.uta.edu/facetedpedia/
[2] http://idir.uta.edu/facetednews/



(a) Facetedpedia interface after selecting one navigational path:
"*American_actors_by_state>New_York_actors>Tom_Cruise*".



(b) Facetedpedia interface after selecting two navigational paths:
"*American_actors_by_state>New_York_actors>Tom_Cruise*" and
"*Film_production_companies_of_Untied_States>Paramount_Pictures*".



(c) Facetedpedia interface after selecting two navigational paths:
"*American_actors_by_state>New_York_actors>Tom_Cruise*" and "*American_film_directors>Steven_Spielberg*".

Fig. 3: Examples of exploring Facetedpedia.

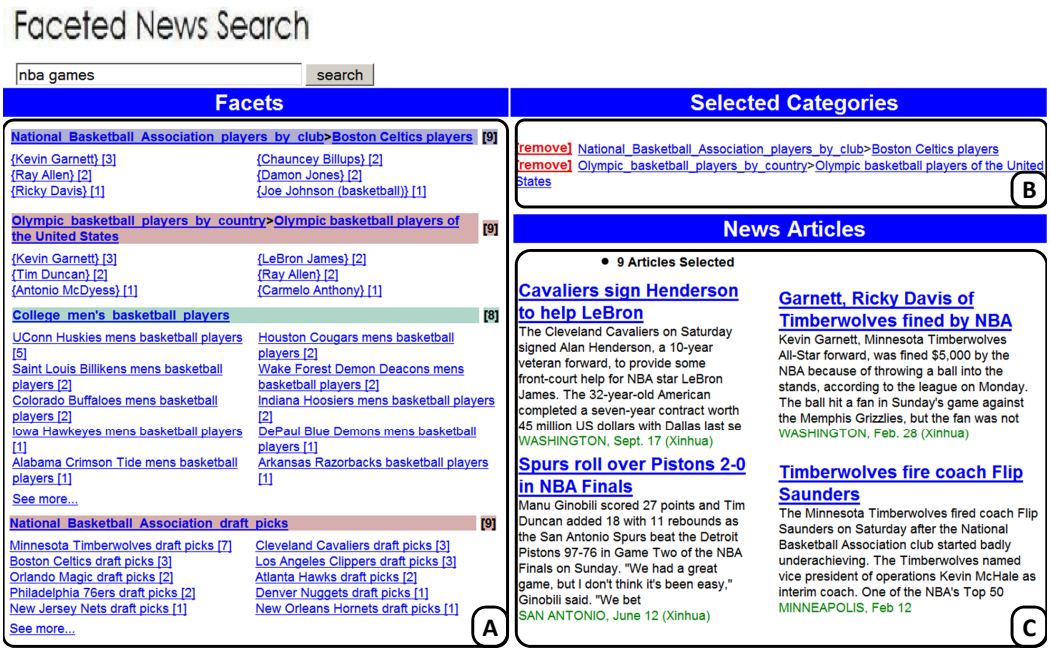


Fig. 4: The faceted search interface of Facetednews.

shows three of the generated facets: (1) American_actors_by_state; (2) American_film_directors; (3) Film_production_companies_of_the_United _States. Other facets include Academy_award_winners, American_film_screenwriters, and so on. Each facet is associated with a hierarchy of categories. Each article can be assigned to the nodes in these hierarchies, with an assignment representing an "attribute" value of the article.

On each facet, the user can navigate through a category path which is formed by parent-child relationships between categories in the Wikipedia category system. [3] [4] *In Figure 3a, the user selects category New_York_actors under American_actors_by_state and she further selects attribute value Tom_Cruise under New_York_actors. A user navigational path is then added in region (B) of Figure 3a. There are four articles satisfying the chosen navigational path, and they are shown in region (C) of Figure 3a. The user could further add another facet condition Paramount_Pictures under category Film_production_companies_of_Untied_States. The result interface is shown in Figure 3b. Here another navigational path is added to region (B) of Figure 3b and the Wikipedia articles covered by both paths are shown in region (C) of Figure 3b.*

If the user is not satisfied with current results, she could remove certain navigational path by clicking "[remove]" in front of that particular path. For example, she could remove the path Film_production_companies_of_Untied_States> Paramount_Pictures and add another path American_ film_directors>Steven_Spielberg. The result interface is shown in Figure 3c. In this way, the user filters the large number of result articles and finds those matching her interests. When the user clicks one particular article title in region (C), the corresponding Wikipedia article would be brought to the user by the system. (This part of the interface is omitted.)

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

[4] The Wikipedia category system is at `http://en.wikipedia.org/wiki/Wikipedia:Categorization`.

Based on the same framework beneath Facetedpedia, Facetednews is a faceted search system for news articles (Figure 4). In a traditional news search engine, a user can search by keywords (e.g. "nba games") and then navigate through the result articles one by one. In Facetednews, several facets relevant to the news articles are generated to help the user narrow down her targets. If she wants to read articles related to Boston Celtics players, she can select category *Boston_Celtics_players* under facet *National_Basketball_Association_players_by_club*. If she is further interested in US players who have played in Olympic games, she can add another facet condition *Olympic_Basketball_players_of_the_United_States*. This scenario is shown is Figure 4. Two navigational paths are added to region (B). She can proceed to read any article in region (C), which contains articles at the intersection of the two chosen navigational paths. She can also add more paths to region (B).

1.2. Overview of Challenges and Solutions

Motivated by the aforementioned examples, we study the problem of dynamic discovery of query-dependent faceted interfaces for text documents. Given a set of top-s ranked articles as the search result from a keyword query, our goal is to produce an interface of multiple facets for exploring these result articles. Specifically, we focus on *automatic* and *dynamic* discovery of faceted interfaces. The facets could not be pre-computed due to the query-dependent nature of our proposed 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. On the contrary, text documents lack such pre-determined 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 rapid growth of document corpora, the large number of attribute values that can be associated with documents, and the complexity of category systems such as the Wikipedia category system, a manual approach would be prohibitively time-consuming and cannot scale to stay up-to-date.

Dynamic discovery of query-dependent faceted interfaces for text documents is a non-trivial undertaking. Below we briefly summarize the main challenges in realizing a faceted search system with such capability and our main ideas:

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

Our 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 documents 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 text documents? and (2) *hierarchy construction* – Where does the category hierarchy of a facet come from?

In this paper we propose a generic model for faceted interfaces, which can be instantiated into different particular systems. Specifically, we instantiate the model into two prototype systems, Facetedpedia and Facetednews, for faceted search over Wikipedia and news articles, respectively. In instantiating the generic model, various sources may be used for identifying facet attributes from text documents. For each document, the named entities appearing in it represent important attributes of that document. Various named entity catalogs can be used in identifying entity mentions in documents. Particularly, a Wikipedia article can be viewed as the description of an entity. Hence Wikipedia itself is a high-quality and comprehensive entity catalog. In addition to treating entities mentioned in articles as facet attributes, we may also use other approaches. For example, if two entities co-occur frequently in a corpus, one can serve as an attribute of the other. This is particularly useful in Facetedpedia, where Wikipedia is both the text corpus and the named entity catalog. Another example is to use categories of Wikipedia articles as their attributes. Furthermore, metadata of articles can also be utilized in forming facet attributes. Once the facet attributes are identified, category hierarchies of facets can be constructed by utilizing various information sources

such as thesaurus (e.g., WordNet [Fellbaum 1998]), taxonomy (e.g., YAGO [Suchanek et al. 2007]), and folksonomy (e.g., Wikipedia category system).

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

Given a set of documents, there may be a large number of candidate facets. Hence a goodness metric for ranking the facets is necessary. Since the ultimate objective of a faceted interface over text documents is to help users explore the documents, it is natural to optimize for finding facets effective for user navigation. A good facet should help users find desirable documents conveniently. Hence we propose to rank facets by their navigational costs, i.e., the amount of effort undertaken by users during navigation, based on a user navigation model.

The problem gets more complex when finding multiple facets to form a faceted interface, 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 text documents, a set of individually "good" facets may not be "good" collectively. Our idea is to avoid overlap between multiple facets and more specifically to optimize for small average pair-wise similarity between facets.

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

A straightforward approach for faceted interface discovery is to enumerate all possible faceted interfaces and apply our ranking metrics directly to find the best interface. Such a brute-force method results in exhaustive examination of all possible combinations of facets. This can easily be a prohibitively large search space. Furthermore, the interactions between the facets in a faceted interface make the computation of its exact ranking score intractable.

Our faceted interface discovery algorithm hinges on two ideas: (1) reducing the search space; and (2) searching the space efficiently. There are two search spaces in finding a good faceted interface– the space of facets and the space of faceted interfaces. To reduce the space of candidate facets, we focus on a subset of "safe reaching facets". To further reduce the space of faceted interfaces, we rank facets individually by their navigational costs and only consider the top ranked facets that do not subsume each other. Instead of exhaustively examining all possible interfaces, we design a hill-climbing based heuristic algorithm that optimizes for both the average navigational cost and the pair-wise similarity of multiple facets.

1.3. Summary of Contributions and Outline

In summary, this paper makes the following contributions:

- *Generic Model of Faceted Interfaces and its Instantiation into Faceted Search Systems over Text Documents.* We propose a generic model of faceted interfaces that is instantiated into two prototype systems, Facetedpedia and Facetednews. To the best of our knowledge, these systems are the first attempt of dynamic discovery of query-dependent faceted interface for text documents. The key idea in our model and instantiation is to exploit collaborative vocabulary (such as Wikipedia category system) as the backbone of faceted interfaces. (Section 3)
- *Metrics for Facet Ranking*. Based on a user navigation model, we propose metrics for measuring the "goodness" of facets, both individually and collectively. (Section 4)
- *Algorithms for Faceted Interface Discovery*. We develop effective and efficient algorithms for discovering faceted interfaces in the large search space of possible interfaces. (Section 5)
- *System Evaluation*. We conducted user study to evaluate the effectiveness of our prototype systems and to compare with alternative approaches. We also quantitatively evaluated their quality and efficiency. (Section 7)

The rest of the paper is organized as follows. In Section 2 we compare Facetedpedia and Facetednews with other systems based on a taxonomy of faceted search systems. In Section 3, we propose the generic model of faceted interface and formally define the various relevant concepts. Section 4

discusses the metrics for ranking facets. We present our facet discovery algorithm in Section 5. Section 6 discusses important implementation details and Section 7 presents the results of user study and experimental evaluation. We review related work in Section 8. Section 9 concludes the paper.

2. FACETED SEARCH SYSTEMS: A COMPARATIVE STUDY

In this section we present taxonomies to characterize relevant faceted search systems and compare them with our work. The systems being compared are those that focus on how to construct faceted interfaces. In Section 8 we will further discuss related works on other aspects of faceted search such as personalization, query log, and user behavior modeling.

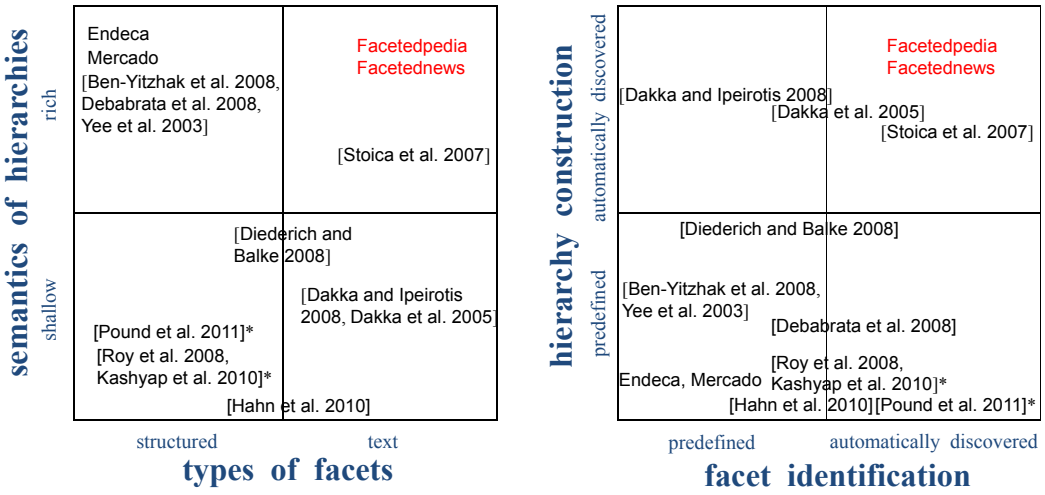
Faceted interface has become influential over the last few years and we have seen an explosive growth of interests in its application [Pollitt 1997; Yee et al. 2003; Hearst 2006; Yee et al. 2003; Hearst 2006; Stoica et al. 2007; Dakka et al. 2005; Dakka and Ipeirotis 2008; Ross and Janevski 2005; Roy et al. 2008; Diederich and Balke 2008; Debabrata et al. 2008; Ben-Yitzhak et al. 2008; Hahn et al. 2010; Kashyap et al. 2010; Pound et al. 2011]. 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 [Pollitt 1997; Hearst 2006; Pratt et al. 1999; Yee et al. 2003; Käki 2005; Pratt et al. 1999; Rodden et al. 2001; Hearst 2006], 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 [Cutting et al. 1992; Zamir and Etzioni 1999; Käki 2005].

To the best of our knowledge, we are the first to propose a query-dependent faceted search framework that discovers both facet dimensions and category hierarchies dynamically. We also demonstrate our framework through two novel application systems: Facetedpedia and Facetednews. Existing research prototypes and commercial faceted search 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.

There exist some previous works on enabling faceted search over Wikipedia. However, they are different from this work, as explained below. CompleteSearch [Bast and Weber 2007] 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, another faceted interface for Wikipedia, Faceted Wikipedia Search [Hahn et al. 2010], has been developed as part of the DBpedia project [Auer et al. 2007]. Their facets are pre-extracted from Wikipedia infobox attributes. Therefore the facets are not dynamically built and hardly query-dependent, as the system often provides the same set of facets for different search result articles. On the contrary, Facetedpedia is fully dynamic and query-dependent. Moreover, it exploits more than 700K Wikipedia categories, in comparison with 1800 pre-extracted attributes from infobox in Faceted Wikipedia Search [Hahn et al. 2010], which makes the facets generated by Facetedpedia more diverse and semantics-rich. Another advantage of our proposed approach is that it can be easily generalized for non-Wikipedia text documents, where infobox does not exist.

Figure 5(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, [Roy et al. 2008; Kashyap et al. 2010; Pound et al. 2011]) or structured attributes in schemata (e.g., [Yee et al. 2003; Debabrata et al. 2008; Ben-Yitzhak et al. 2008]) 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., [Stoica et al. 2007]) or frequency-based subsumption relationships between general and specific terms (e.g., [Dakka et al. 2005; Dakka and Ipeirotis 2008]).



(a) Facet types and semantics.

(b) Automation and dynamism.

Fig. 5: Taxonomies of faceted search systems. (Works marked by * do not support category hierarchy on facet.)

These systems cannot leverage as much semantic information. The work [Diederich and Balke 2008] is in the middle of Figure 5(a) since it has both structured dimensions and a subsumption-based topic taxonomy. In Faceted Wikipedia Search [Hahn et al. 2010] the facets are from metadata (i.e., attributes in Wikipedia infoboxes) of its target data. Hence it is in the middle along the dimension of type of facets.

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

Figure 5(b): Taxonomy by Degree of Automation and Dynamism

When building the two pillars in a faceted interface, namely the facet and the hierarchy, our framework is both automatic and dynamic, as motivated in Section 1.2. On this aspect, none of the existing systems could be effectively applied, because none is fully automatic in both facet identification and hierarchy construction.

In some systems (e.g., Endeca, Mercado, [Roy et al. 2008; Ben-Yitzhak et al. 2008; Yee et al. 2003; Debabrata et al. 2008; Kashyap et al. 2010]) the dimensions and hierarchies are predefined, therefore they do not discover the facets or construct the hierarchy. In [Debabrata et al. 2008; Roy et al. 2008; Kashyap et al. 2010; Hahn et al. 2010] a subset of interesting/important facets are automatically selected from the predefined ones. In [Dakka et al. 2005; Dakka and Ipeirotis 2008] the set of facets are predefined, but the hierarchies are automatically created based on subsumption. In [Diederich and Balke 2008] only one special facet (a topic taxonomy) is automatically generated and the rest are predefined. In [Pound et al. 2011] facets are automatically discovered, but it does not use hierarchies on facets.

With respect to the automation of faceted interface discovery, the closest work to ours is the Castanet algorithm [Stoica et al. 2007]. 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.

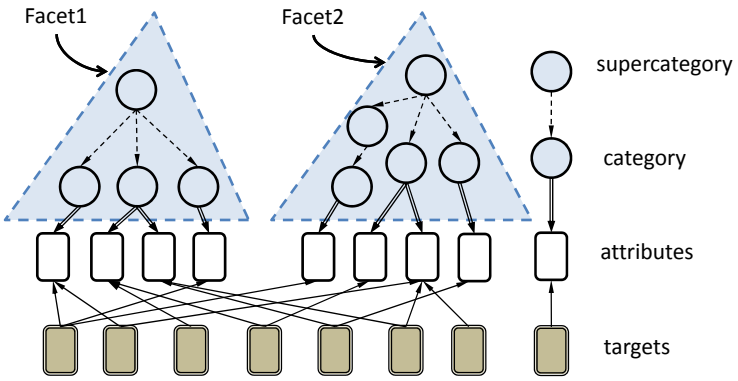


Fig. 6: The generic model for faceted interfaces.

3. A GENERIC MODEL FOR FACETED INTERFACES AND ITS INSTANTIATION FOR FACETED SEARCH OVER TEXT DOCUMENTS

In this section we first present a generic model of faceted interfaces, to explain the basic concepts in faceted search systems. We then discuss how to instantiate the model to Facetedpedia and Facetednews, which are our faceted search systems for Wikipedia and news articles, respectively. Finally we formally define the concepts in this model.

3.1. A Generic Model of Faceted Interfaces

Figure 6 is a generic model that shows the basic components in faceted interfaces and their relationships. Most faceted search interfaces consist of three levels: targets, attributes, and category hierarchies. The *targets* are the objects that users browse and search for. Examples of targets include database records, merchandise, photos, videos, web bookmarks, library collections, news articles, and Wikipedia entities (e.g., people, places, organizations, etc.), depending on application scenarios. The *attributes* are the features of the targets. Examples include database schema attributes, product features (e.g., *price*, *manufacturer*, *size*, etc.), tags on social media objects such as photos, videos, and bookmarks, metadata of library collections, terms in articles, and named entities related to target entities. The attributes of objects are partitioned into multiple *facets*, each of which corresponds to a dimension of attributes for exploring the objects. In some systems, each facet is simply a flat group of attribute values. In other systems, the attribute values in each facet can be further organized into a *category hierarchy*, which ideally presents the IS-A relationships between category-subcategory and category-attribute. Various ways can be exploited in constructing category hierarchy. For instance, a category hierarchy can be derived from a folksonomy such as the Wikipedia category system, a thesaurus such as WordNet, or a domain-specific taxonomy (e.g., the city-state-country hierarchy for places).

Figure 7 shows several possible instantiations of the above generic model for building faceted interfaces in various applications. Note that the figure is only for illustration purpose. It is not meant to cover all possible scenarios of instantiation. For each scenario in the figure, we show what are the targets, the attributes, and the categories, to instantiate the model. Figure 7(a) shows that each facet over a relational database table corresponds to an attribute in its schema. Attribute values in a facet can be hierarchically organized by a domain-specific taxonomy or simply alphabetically ordered. The same instantiation can be used for faceted search over products in online stores and museum and library collections. Figure 7 (b) shows how to build a faceted interface for text documents based on the generic model. The attributes are named entities appearing in target documents and/or related to the documents. The categories of these named entities and the super-categories of the categories form the category hierarchies of facets. These hierarchies are based on a folksonomy

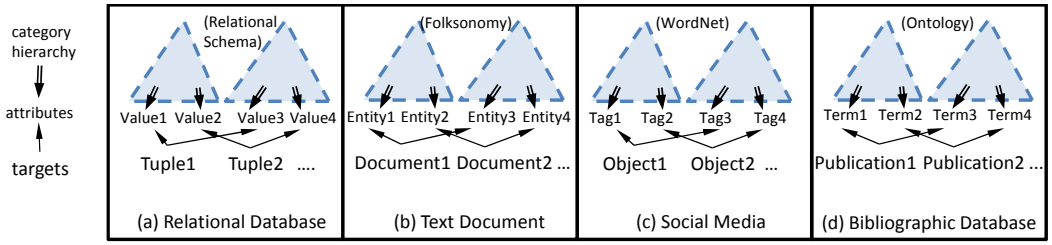


Fig. 7: Instantiations of the generic faceted interface model for different scenarios.

such as the Wikipedia category system. The details of this instantiation are given in Section 3.2. Figure 7(c) is an instantiation of the model for faceted search over social media objects (e.g., videos and photos) and social bookmarks. Each tag is an attribute. Related tags are put together as a facet and hierarchically organized by concepts from a thesaurus taxonomy such as WordNet [Fellbaum 1998]. By choices made on multiple facets, a user can find those objects that have the specified tags or have tags belonging to the chosen concepts. Figure 7(d) shows how to instantiate the model for a faceted interface over a bibliographic database (e.g., PubMed [5]). The attributes are the scientific terms appearing in target objects, i.e., publications. The terms are organized into multiple facets with hierarchical categories, based on specialized taxonomies such as the Gene Ontology [6].

3.2. Instantiation of the Generic Model for Faceted Interfaces over Text Documents

In this section we provide detailed discussion of Figure 7(b), i.e., how to instantiate the generic model for faceted search over text documents. Specifically, we explain its instantiation into two prototype faceted search systems, Facetedpedia and Facetednews, for Wikipedia articles and news articles, respectively.

The basic components in the generic model are targets, attributes, and category hierarchies. Hence the key to realization of the model is to instantiate these basic components. 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 attributes (cf. Figure 7(a)). On the contrary, text documents lack such predetermined schemata. To address this challenge, the basis of our instantiation is to exploit user-generated *collaborative vocabulary* in Wikipedia such as its "grassroots" category system and its heavily interlinked articles. The collaborative vocabulary represents the collective intelligence of many users and rich semantic information, and thus constitutes the promising basis for forming faceted interfaces. With regard to the concept of facet attribute, the Wikipedia articles (i.e., entities) hyperlinked from or related to a search result article (i.e. target article) are exploited as its attributes. 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 concepts during exploration.

In Facetedpedia, the targets are Wikipedia articles, each of which is an encyclopedia entry describing the corresponding entity. For each target entity, its attributes are also Wikipedia entities. They co-occur frequently with the target entity within syntactical units such as sentences or paragraphs in some text corpus. The premise is that if an entity co-occurs frequently with the target entity, there is a good chance that they are highly related. Specifically we use Wikipedia itself as the text corpus for measuring degree of co-occurrence, although other text corpora can also serve the purpose, after Wikipedia entities appearing in documents are annotated. The attribute entities can also be identified by multiple ways together. For example, attribute entities of a target entity can be the hyperlinked entities in the target. In Wikipedia, the fact that the authors of an article (target

[5] http://www.ncbi.nlm.nih.gov/pubmed/
[6] http://www.geneontology.org/

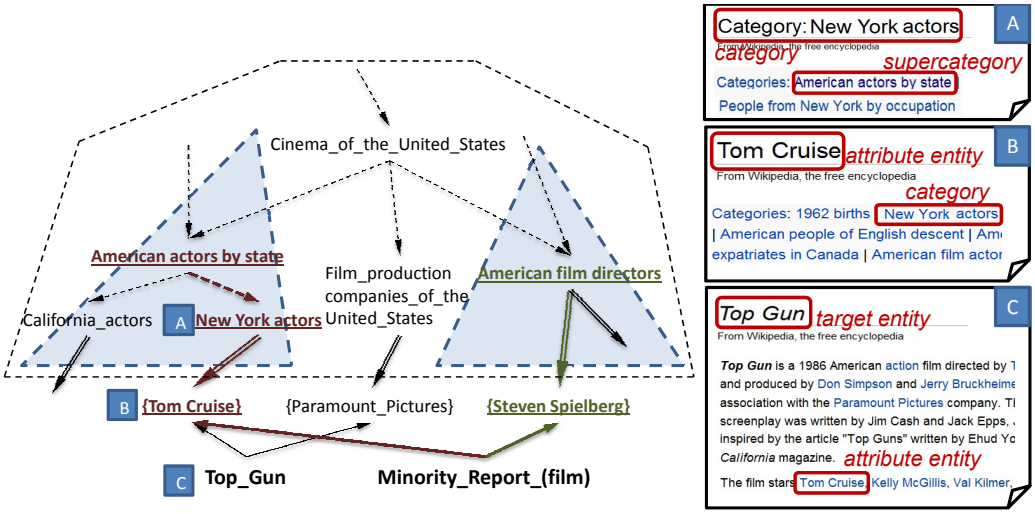


Fig. 8: Instantiation of the generic model for Facetedpedia. Two highlighted navigational paths corresponding to Figure 3c.

entity) collaboratively made hyperlinks to other entities is an indication of the significance of the hyperlinked entities in describing the target entity.

In Facetednews, the targets are news articles. Similar to Facetedpedia, Facetednews also uses Wikipedia entities as attributes of target articles. Specifically, given a target article, its attribute entities are mentioned in the article itself. The attribute entities are recognized by entity annotation techniques and particularly entity annotation systems developed for annotating by Wikipedia. We use Wikifier[7] [Milne and Witten 2008] for such purpose. Given a text document, Wikifier adds hyperlinks to Wikipedia entities in the document. It does so by matching token sequences in the document with titles of Wikipedia entities and disambiguating among candidate matches. After annotation, the news articles in Facetednews are enriched with semantic attributes that are helpful in faceted navigation. This instantiation is applicable to not only news articles but also general text documents, which extends the application scenarios of our generic model.

Since both Facetedpedia and Facetednews use Wikipedia entities as attributes of target articles, the instantiation of category hierarchies is the same in the two systems, by exploiting the Wikipedia category system. [8] The Wikipedia category system consists of a large hierarchy of categories. Its root is Category:Fundamental categories. [9] Starting from the root, a category may contain multiple subcategories and multiple instance articles. Subcategories cover Wikipedia articles under more specific concepts, while supercategories cover more general concepts. Articles contained in a category and its subcategories are instances of the concept covered by the category. Hence the Wikipedia category system is a hierarchy formed by supercategory-subcategory relationships and category-instance article relationships.

In both Facetedpedia and Facetednews, each facet is a sub-hierarchy of the Wikipedia category system. A facet groups together highly-related attribute entities (which are Wikipedia articles themselves). Therefore a facet corresponds to a conceptual dimension and each attribute entity in it corresponds to a value on the dimension. The hierarchy of categories in the facet helps users to

[7] http://www.nzdl.org/wikification/docs.html

[8] Note that the generic model of faceted interfaces is not limited to the specific Wikipedia category system. Ideally, any category hierarchy from a well-structured taxonomy such as WordNet [Fellbaum 1998] or YAGO [Suchanek et al. 2007] can be exploited as the category hierarchy in the model.

[9] http://en.wikipedia.org/wiki/Category:Fundamental_categories

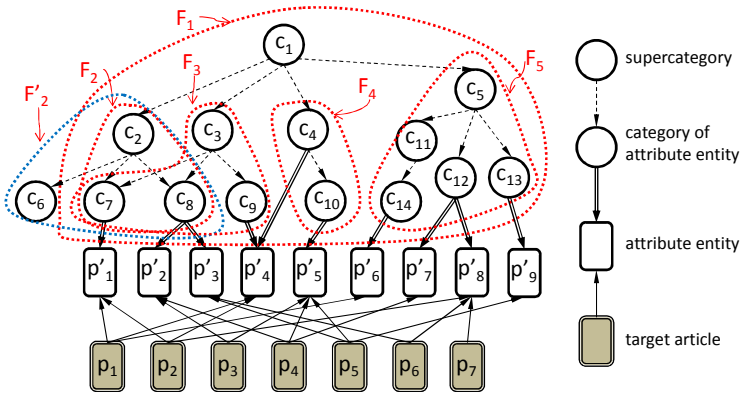


Fig. 9: The concept of facet for documents.

explore from general categories to more specific ones, then to instance articles of categories (i.e., attribute entities), and finally to target entities that attain the attribute entities.

Example 2 (Instantiate the Generic Model for Facetedpedia) *Figure 8 shows an example of instantiating the generic model into Facetedpedia. The target entities are two Wikipedia articles Top_Gun and Minority_Report_(film). The attribute entities are Wikipedia articles Tom_Cruise, Paramount_Pictures, and Steven_Spielberg. The two facets are sub-hierarchies under Wikipedia categories American_actors_by_state and American_film_directors. The three boxes on the right side of the figure represent Wikipedia pages A, B, and C. In this particular example, we consider the hyperlinked entities on a target article as the attribute entities of that target. Hence Tom_Cruise is an attribute entity of target entity Top_Gun, according to Wikipedia page C, which is the article for the target entity itself. New_York_actors is a category of attribute entity Tom_Cruise, according to Wikipedia page B, which is the article for the attribute entity itself. American_actors_by_state is a supercategory of New_York_actors, by Wikipedia page A.*

3.3. Definitions of Concepts and Faceted Interface Discovery Problem

We now formally define the concepts of faceted interfaces for text documents, based on the aforementioned instantiation of our generic model. We also provide the specification of faceted interface discovery problem.

Definition 1 (Target Article, Attribute Entity) *Given a keyword query q, the set of top-s ranked result articles from search engine, $\mathcal{T}=\{p_1,...,p_s\}$, are the* target articles *of q. Each target article can have multiple* attribute entities*, each of which is a Wikipedia article (entity). The relationship between a target article p and its attribute entity p' is represented as $p \rightarrow p'$. The relationship can be established by different measures, such as p and p' co-occurring enough number of times (when p itself is also a Wikipedia entity), p' is mentioned in article p or simply hyperlinked from p, and so on. Given $\mathcal{T}$, the set of attribute entities is $\mathcal{A}=\{p'_1,...,p'_m\}$, where each p'_i is an attribute entity of at least one target article $p_j \in \mathcal{T}$.*

Definition 2 (Category Hierarchy) *A* 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 a set of categories, the edge set $\mathcal{E}_\mathcal{H}=\{c \dashrightarrow c'\}$ is a set of supercategory(c)-subcategory(c') relationships, and $r_\mathcal{H}$ is the root of $\mathcal{H}$.*

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 3 (Running Example) *In Figure 9 there are* 7 *target articles (*p_1*, ..., *p_7*) and* 9 *attribute entities (*p'_1*, ..., *p'_9*). The category hierarchy has* 14 *categories (*c_1*, ..., *c_{14}*). The figure highlights* 6 *facets (*$\mathcal{F}_1$*, ..., *$\mathcal{F}_5$*, 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 a real category hierarchy of a folksonomy such as Wikipedia category hierarchy. We will use Figure 9 as the running example throughout the paper.*

The categories in a facet can "*reach*" target articles $\mathcal{T}$ through attribute entities $\mathcal{A}$. That is, by following the category-subcategory hierarchy of the facet, we can find a category, then find an attribute entity belonging to the category, and finally find the target articles that have the attribute. All such target articles are called the *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 target articles* $\mathcal{T}$*, attribute entities* $\mathcal{A}$*, 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* $1 \leq i \leq t$*,* $c_i \in \mathcal{C}_\mathcal{F}$*, i.e.,* c_i *is a category in* $\mathcal{F}$*;*
- *for* $1 \leq i \leq t-1$*,* $c_i \dashrightarrow c_{i+1} \in \mathcal{E}_\mathcal{F}$*, 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 entity of* p *(represented as* $p \rightarrow p'$*).*

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

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

Example 4 (Navigational Path and Faceted Interface) *Continue the running example. In Figure 9,* $I = \{\mathcal{F}_2, \mathcal{F}_5\}$ *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** over text documents is: Given a 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 entities $\mathcal{A}$, 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 articles to a keyword query, $\mathcal{T}$, there are a large number of attribute entities 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. 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.

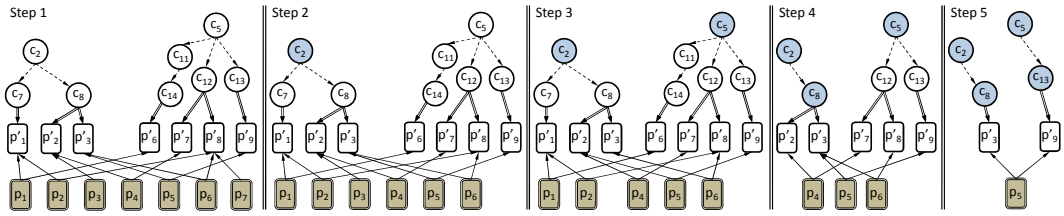


Fig. 10: The navigation on a 2-facet interface $\mathcal{I} = \{\mathcal{F}_2, \mathcal{F}_5\}$.

Given that faceted interfaces are for users to navigate through the associated category hierarchies and to ultimately reach the target articles, it is natural to rank them by the users' navigational cost, i.e., the amount of effort undertaken by the users 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 in multiple facets in a k-facet interface. At the beginning, the navigation starts from the roots of all 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 entities reachable from the current category. The selections made on the k facets together form a conjunctive query. After the selection at each step, the list of target articles that satisfy the conjunctive query are brought to the user. The navigation terminates when the user decides that she has seen desirable target articles.

Example 5 (Navigation in Faceted Interface) *Continue the running example in Figure 9. Consider a faceted interface I=$\{\mathcal{F}_2, \mathcal{F}_5\}$. A sequence of navigational steps on this interface are in Figure 10. 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 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 cost of an individual facet. 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, has a small number of steps and at each step only requires the user to browse a small number of choices. Therefore, we formally define the cost of a navigational path as the summation of fan-outs (i.e., number of choices) at every step, in logarithmic form. [10]

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

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

[10] 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 procedure similar to binary search.

where l=c_1 --→...--→c_t⇒p'←p.

In Formula 1, $fanout(p')$ *is the number of (directly) reachable target articles through the attribute entity* p'*,*

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

$$\mathcal{T}_{p'} = \{p | p \in \mathcal{T} \wedge p \rightarrow p'\} \tag{3}$$

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

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

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

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

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

$$\mathcal{C}_c = \{c' | c' \in \mathcal{C}_\mathcal{F} \wedge c \dashrightarrow c' \in \mathcal{E}_\mathcal{F}\} \tag{6}$$

Note that we made several assumptions for simplicity. The cost formula only captures "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 a user decides to change a previous selection. Furthermore, we assume the user always completes a navigational path till reaching 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 6 (Cost of Navigational Path) *We continue the running example. Given* l= c_5 --→ c_{12}⇒p'_8←p_6*, a navigational path of* $\mathcal{F}_5$ *in Figure 9,* $cost(l)$=$fanout(c_5)$+$fanout(c_{12})$+ $fanout(p'_8)$=$log_2(3)$+$log_2(2)$+$log_2(3)$=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 in $\mathcal{T}$, 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 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* $penalty$*.*

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

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

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

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

$$l_p = \{l | l = r \dashrightarrow ... \Rightarrow p' \leftarrow p\} \tag{9}$$

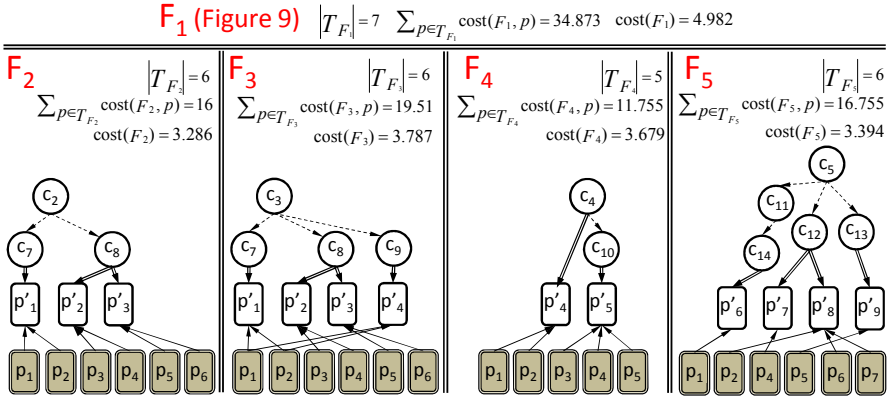


Fig. 11: Navigational costs of facets.

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 7) and is at least larger than the highest cost of any path to a reachable target article.

Example 7 (Cost of Facet) *We continue the running example. Figure 11 shows the costs of the* 5 *highlighted facets in Figure 9, together with their category hierarchies and reachable attribute entities and target articles. It does not show* $\mathcal{F}_1$ *which is Figure 9 itself excluding* c_6*. The costs of facets are obtained by Formula 7, with* $penalty$=7*. For instance,* $cost(\mathcal{F}_2)=\frac{1}{7}\times(\sum_{p\in\{p_1,p_2,p_3,p_4,p_5,p_6\}} cost(\mathcal{F}_2,p)+penalty\times|\mathcal{T}-\mathcal{T}_{\mathcal{F}_2}|)=\frac{1}{7}\times(16+7\times1)=3.286$*.* $\mathcal{F}_2$ *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.*

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 nontrivial. This is because the best k-facet interface may not be simply the set of cheapest k facets. The reason is that when a user navigates multiple facets, the selection made at one facet has impact on the available choices on other facets, as illustrated by Example 5.

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 9. Two possible sequences of navigational steps are shown in Figure 12(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 10, 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 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 a faceted interface. The interaction between facets is query- and data-dependent, rendering such exhaustive computation practically infeasible.

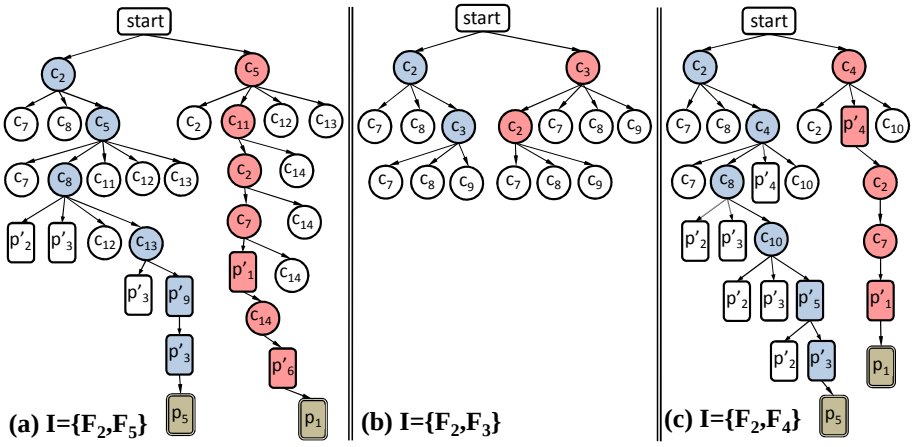


Fig. 12: The sequences of navigational steps.

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 12(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 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 entities, defined below.

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

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

where the similarity between two facets $sim(\mathcal{F}_i, \mathcal{F}_j)$ is defined by an extension of overlap coefficient *[Rijsbergen 1979],*

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

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 entities reachable from $\mathcal{F}_i$,

$$\mathcal{A}_{\mathcal{F}_i} = \{p' | p' \in \mathcal{A} \wedge \exists c \in \mathcal{C}_{\mathcal{F}_i} \; s.t. \; c \Rightarrow p'\} \tag{12}$$

Example 8 (Similarity of Facets) *Consider facets $\mathcal{F}_1$,...,$\mathcal{F}_5$ in Figure 9.* $sim(\mathcal{F}_2, \mathcal{F}_3) = \frac{|\mathcal{C}_{\mathcal{F}_2} \bigcap \mathcal{C}_{\mathcal{F}_3}| + |\mathcal{A}_{\mathcal{F}_2} \bigcap \mathcal{A}_{\mathcal{F}_3}|}{\min(|\mathcal{C}_{\mathcal{F}_2}|, |\mathcal{C}_{\mathcal{F}_3}|) + \min(|\mathcal{A}_{\mathcal{F}_2}|, |\mathcal{A}_{\mathcal{F}_3}|)} = \frac{|\{c_7, c_8\}| + |\{p'_1, p'_2, p'_3\}|}{\min(|\{c_2, c_7, c_8\}|, |\{c_3, c_7, c_8, c_9\}|) + \min(|\{p'_1, p'_2, p'_3\}|, |\{p'_1, p'_2, p'_3, p'_4\}|)} = 5/6.$
Other pari-wise facet similarities are computed in the same way. The average pari-wise similarity of faceted interface $\mathcal{I}$=$\{\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_5\}$ is $sim(\mathcal{I}) = (sim(\mathcal{F}_2, \mathcal{F}_3) + sim(\mathcal{F}_2, \mathcal{F}_5) + sim(\mathcal{F}_3, \mathcal{F}_5))/3 = 5/18.$

For multi-facet ranking, we do not design a single function to combine the average pair-wise similarity of facets in a faceted interface with its navigational cost, since the measures of similarity and navigational cost are of 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 category hierarchy $\mathcal{H}$ and apply our ranking metrics directly to find the best interface. Such a brute-force 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 the category system in Wikipedia. The brute-force 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.

Based on these ideas, our k-facet discovery algorithm consists of 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 category hierarchy $\mathcal{H}(r_{\mathcal{H}}, \mathcal{C}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$, target articles $\mathcal{T}$, and attribute entities $\mathcal{A}$, 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 entity $p' \in \mathcal{A}$ or a super-category 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 can 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. Hence we 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 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*

Algorithm 1: Construct RCH and Get Attribute Entities

Input: $\mathcal{T}$: target articles; $\mathcal{H}$: category hierarchy.
Output: $\mathcal{A}$:attribute entities; $\mathcal{RCH}$:relevant category hierarchy.

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

$\mathcal{F}$ all the descendants of the root r and their category-subcategory relationships are retained from $\mathcal{RCH}$.

Example 9 ($\mathcal{RCH}$ and $\mathcal{RCH}$-Induced Facet) *Continue the running example. In Figure 9, 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 the searching for k-facet interfaces are performed on it.

5.2. Ranking Single Facet (Algorithm 2 and 3)

Among all $\mathcal{RCH}$-induced facets, only top n facets with the smallest navigational costs are considered in searching for a k-facet interface ($k < n$). In ranking facets by their costs, one straightforward approach is to enumerate all $\mathcal{RCH}$-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 $\mathcal{RCH}$-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 $\mathcal{RCH}$-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, during the recursive traversal of $\mathcal{RCH}$, record 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 recorded information of the facets rooted at r's direct subcategories, without accumulating the individual costs

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

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 entities belonging to r.
8 $fanout(r) \leftarrow |\mathcal{A}_r| + |\mathcal{C}_r|$
9 $\mathcal{T}_r \leftarrow (\cup_{p' \in \mathcal{A}_r} \mathcal{T}_{p'}) \bigcup (\cup_{c \in \mathcal{C}_r} \mathcal{T}_c)$ // reachable target articles of r.
10 **foreach** $p \in \mathcal{T}_r$ **do**
11 $pathcnt(\mathcal{F}_r, p) \leftarrow |\{p' | p' \in \mathcal{A}_r, p \in \mathcal{T}_{p'}\}| + \sum_{c \in \mathcal{C}_r} pathcnt(\mathcal{F}_c, p)$
12 $cost_1 \leftarrow \sum_{p' \in \mathcal{A}_r s.t. p \in \mathcal{T}_{p'}} (log_2(fanout(r)) + log_2(fanout(p')))$
13 $cost_2 \leftarrow \sum_{c \in \mathcal{C}_r} (log_2(fanout(r)) + cost(\mathcal{F}_c, p)) \times pathcnt(\mathcal{F}_c, p)$
14 $cost(\mathcal{F}_r, p) \leftarrow \frac{cost_1 + cost_2}{pathcnt(\mathcal{F}_r, p)}$
15 $cost(\mathcal{F}_r) \leftarrow \sum_{p \in \mathcal{T}_r} cost(\mathcal{F}_r, p) + penalty \times |\mathcal{T} - \mathcal{T}_r|$
16 **return**

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.

Algorithm 4: k-Facet Interface Selection

Input: $\mathcal{I}_n$: the top n $\mathcal{RCH}$-induced facets with the smallest costs.
Output: $\mathcal{I}_k$: a discovered faceted interface with k facets ($k<n$).

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

// hill climbing
2 $\mathcal{I}_k \leftarrow$ a random k-facet subset of $\mathcal{I}_{n^-}$; $\mathcal{I}' \leftarrow \mathcal{I}_{n^-} \backslash \mathcal{I}_k$
3 **repeat**
5 make $\mathcal{I}_k$=<$\mathcal{I}_k[1]$,...,$\mathcal{I}_k[k]$> sorted in increasing order of cost.
6 make $\mathcal{I}'$=<$\mathcal{I}'[1]$,...,$\mathcal{I}'[n-k]$> sorted in increasing order of cost
7 **for** $i = k$ **to** 1 **step** -1 **do**
8 **for** $j = 1$ **to** $n-k$ **do**
9 $\mathcal{I}_{new} \leftarrow (\mathcal{I}_k \backslash \{\mathcal{I}_k[i]\}) \cup \{\mathcal{I}'[j]\}$
10 $S_1 \leftarrow \sum_{\mathcal{F}_c, \mathcal{F}_{c'} \in \mathcal{I}_{new}, \mathcal{F}_c \neq \mathcal{F}_{c'}} sim(\mathcal{F}_c, \mathcal{F}_{c'})$
11 $C_1 \leftarrow \sum_{\mathcal{F}_c \in \mathcal{I}_{new}} cost(\mathcal{F}_c)$
12 $S_2 \leftarrow \sum_{\mathcal{F}_c, \mathcal{F}_{c'} \in \mathcal{I}_k, \mathcal{F}_c \neq \mathcal{F}_{c'}} sim(\mathcal{F}_c, \mathcal{F}_{c'})$
13 $C_2 \leftarrow \sum_{\mathcal{F}_c \in \mathcal{I}_k} cost(\mathcal{F}_c)$
14 **if** *(*$S_1 \leq S_2$ **and** $C_1 < C_2$*)* **or** *(*$S_1 < S_2$ **and** $C_1 \leq C_2$*)* **then**
15 $\mathcal{I}_k \leftarrow \mathcal{I}_{new}$; $\mathcal{I}' \leftarrow \mathcal{I}_{n^-} \backslash \mathcal{I}_k$
16 go to line 5
17 **until** $\mathcal{I}_k$ *does not change*;
18 **return** $\mathcal{I}_k$

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 $\mathcal{I}_{n^-}$, 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. SYSTEM IMPLEMENTATION

The generic model (Section 3), ranking metrics (Section 4), and algorithms (Section 5) are instantiated into two prototype systems: Facetedpedia and Facetednews. Below we introduce the implementation details of both systems.

6.1. Facetedpedia

In Facetedpedia, the targets are Wikipedia entities (i.e., articles). The attributes of a target are Wikipedia entities that co-occur at least twice with the target entity in the text corpus– Wikipedia

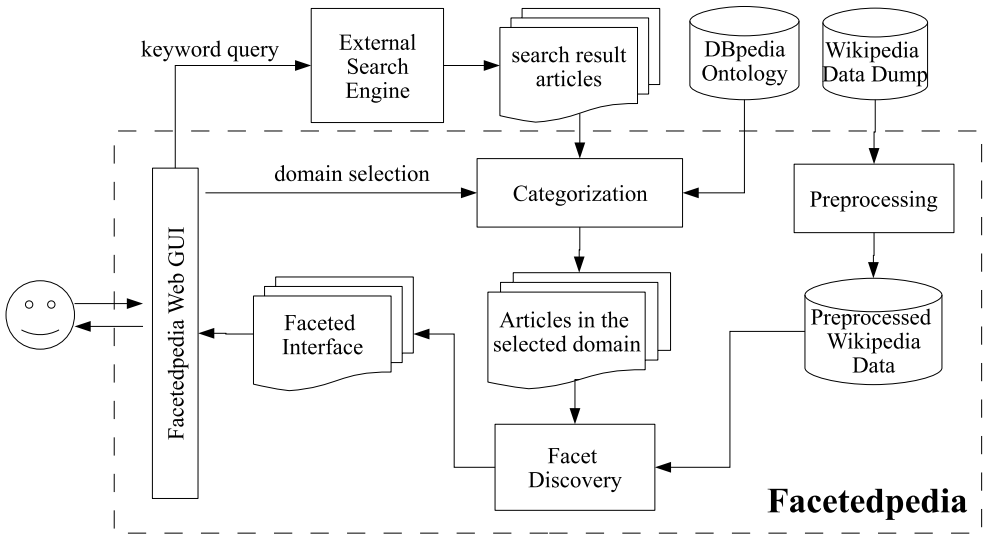


Fig. 13: The architecture of Facetedpedia.

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

Fig. 14: Characteristics of the Wikipedia dataset.

itself. The Facetedpedia system mainly consists of four components: preprocessing Wikipedia data dump, categorization, facet discovery, and Facetedpedia web GUI. The architecture of the system is shown in Figure 13. We further elaborate the implementation of the four components as follows.

Preprocessing Wikipedia Data Dump: We used the Wikimedia MySQL data dump generated on July 24th 2008[11] and loaded the data into our local 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 these tables. One major preprocessing task is to clean the original category hierarchy to make it cycle-free. Although cycles should usually be avoided as suggested by Wikipedia, the category system in Wikipedia contains a small number of elementary cycles [12] (594 detected in the dataset) due to various reasons. We applied depth-first search algorithm to detect elementary cycles in the original dataset. The category hierarchy is made acyclic by removing the last encountered edge in each elementary cycle during the depth-first search. Other 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

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

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

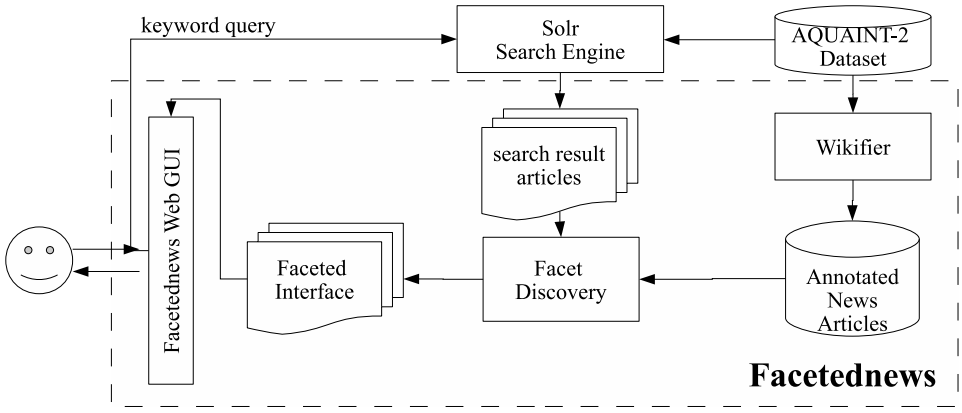


Fig. 15: The architecture of Facetednews.

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 14. The total size of the tables is 1.2GB.

Categorization: A faceted interface is more effective on a set of homogeneous target entities. In our implementation, we exploited the DBpedia ontology[13] for assigning Wikipedia entities to about 80 pre-determined domains (e.g., People, Places, etc). This is done offline and the categorization result of all Wikipedia entities is stored in a database table.

When a user issues a keyword search query, the query is sent to an external search engine which returns a ranked list of Wikipedia entities (i.e., articles). The returned entities are most likely from different domains, thus Facetedpedia asks the user to select one particular domain of her interest. An alternative approach is to let Facetedpedia select the dominant or largest domain of result entities automatically. However, the user may like to select from other domains of her interest. A k-facet interface is then discovered for the top-s search result entities belonging to the chosen domain. In our implementation, we use Google.com as the external search engine.

Facet Discovery: The facet discovery component is a multi-thread background daemon program. The main process creates a new thread for each user session. The main process pre-loads all the preprocessed tables (1.2GB in total) into memory. After the user chooses a target domain, a new thread is created to run the facet ranking and search algorithms and generate the resulting faceted interface.

Facetedpedia web GUI: The generated faceted interface, including information such as the category hierarchy of each facet and the entities reachable from each category in the hierarchy, is stored in a database. The GUI is a dynamic web page implemented using Ajax. It reads the generated interface data from the database, displays the faceted interface, and updates the interface based on the user's navigation.

6.2. Facetednews

In Facetednews, the targets are news articles and the attributes are Wikipedia entities, as mentioned in Section 3. Facetednews mainly consists of three components: preprocessing news data, facet discovery, and Facetednews web GUI. The system architecture is shown in Figure 15. In Facetednews we do not categorize news articles. We did not find it empirically important to require news arti-

[13] http://wiki.dbpedia.org/Ontology

cles to be "homogeneous" in order to make a faceted interface over the news articles effective. The implementation of facet discovery and web GUI components in Facetednews is similar to that in Facetedpedia. In order to apply our faceted interface discovery algorithm to news articles, we implemented two additional functionalities to preprocess data– indexing and annotating news articles, which we further elaborate below.

Indexing news articles using Apache Solr: We used AQUAINT-2 dataset[14] as our news corpus. It consists of 907K news articles from 6 news agencies in the period of October 2004 – March 2006. We index the news articles and provide full-text search over the articles, by using the Apache Solr full-text search engine [15]. For a keyword query to the Facetednews web GUI, the search system returns a list of news articles, which become the target articles for facet discovery. We did not use a commercial news search engine to fetch news articles for queries, to avoid the overhead of query-time news article extraction and annotation. That way we can stay focused on our objective of investigating how to construct faceted interfaces over news articles.

Annotating news articles using Wikifier: We applied Wikifier [Milne and Witten 2008] to annotate news articles, i.e., to identify Wikipedia entity names mentioned in the articles. For discovering faceted interfaces over the news articles, the identified entities are their attribute entities. Over the 907K news articles in AQUAINT-2 corpus, Wikifier detected over 13 million attribute entities, i.e., in average 14 attribute entities for each news article.

7. EVALUATION

Our experiment was conducted 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.

7.1. User Studies

We conducted user studies [16] to evaluate the effectiveness of both Facetedpedia and Facetednews. For faceted interface discovery over Wikipedia articles, we compared the results generated from three systems: Facetedpedia, Castanet [Stoica et al. 2007], and Faceted Wikipedia Search [Hahn et al. 2010]. 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 dynamic keyword search results over Wikipedia. We used the same graphical user interface for both systems to make the comparison independent from interface design difference. As for Faceted Wikipedia Search, we do not have the implementation of its internal algorithms. Thus, we utilized its online service for our user studies. This might lead to biases in system preference due to different GUI design. For faceted interface discovery over news articles, we compared the results generated from Facetedpedia and Castanet. As explained in Section 2, Faceted Wikipedia Search is inapplicable for faceted interfaces over general text documents, since it uses Wikipedia infoboxes for generating facets.

In Facetedpedia, each query is sent to Google with site constraint *site:en.wikipedia.org* to get the top 200 (s=200) English Wikipedia articles. In Facetednews, each query is sent to our local Solr search engine to get the top 400 (s=400) news articles. The relevant category hierarchy ($\mathcal{RCH}$) is then generated by applying Algorithm 1 on the aforementioned MySQL database. By default, Algorithm 2 (facet ranking) returns top 200 (n=200) facets and Algorithm 4 (faceted interface selection) generates 20 facets (k=20). The value of $penalty$ in Definition 7 was empirically selected by investigating the relationship between number of unreachable target articles ($|\mathcal{T} - \mathcal{T}_r|$) and the total navigational cost of reachable targets ($\sum_{p \in \mathcal{T}_r} cost(\mathcal{F}_r, p)$).

---

[14] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T25

[15] http://lucene.apache.org/solr/

[16] All the survey pages we used for our user studies are provided at http://idir.uta.edu/facetsurvey/.

Query ID	Facetedpedia	Query ID	Facetednews
1	us action film	1	nba
2	us national park	2	ford
3	us country singer	3	mobile phone
4	album	4	pc game
5	us film star	5	layoff
6	pc game	6	bankruptcy
7	computer scientist	7	tsunami
8	football player	8	president election
9	software	9	microsoft
10	best seller book	10	asia market
11	american writer	11	texas university
12	interstate highway	12	terrorism

Fig. 16: Queries.

us action film	computer scientist
American_actors_by_state	Association_of_American_Universities
American_writers	Liberal_democracies
American_film_directors	Software_companies_of_the_United_States
Academy_Awards	Companies_in_the_NASDAQ-100_Index
Liberal_democracies	Host_cities_of_the_Summer_Olympic_Games
Expatriates_in_the_United_States	History_of_human-computer_interaction
Film_directors_by_genre	Xerox
Film_score_composers_by_nationality	!!!_albums
Fictional_secret_agents_and_spies	Computer_hardware_companies
English_film_actors	Companies_established_in_the_1980s
us country singer	**best seller book**
Albums_by_year	American_writers_by_genre
Liberal_democracies	American_writers_by_state
Singles_by_year	English-language_films
Radio_formats	American_military_personnel_of_World_War_II
American_record_labels	Faculty_by_university_or_college_in_the_United_States
Companies_based_in_New_York_City	People_by_high_school_in_the_United_States
Nashville,_Tennessee	People_of_English_descent
2000s_films	American_actors_by_state
Spoken_articles	Agnostics_by_nationality
County_seats_in_Tennessee	Screenwriters_by_nationality

Fig. 17: Root categories of 10 facets in the faceted interfaces generated by Facetedpedia for 4 queries.

Both Factedpedia and Facetednews were evaluated for 12 keyword queries, listed in Figure 16. For Facetedpedia, we made the query keywords distributed across different domains. For Facetednews, we chose keywords based on news events during the corresponding period of the news corpus. Figure 17 shows a sample of resulting facets produced by Facetedpedia for 4 of the queries. For each query, it shows the root categories of 10 facets in the faceted interface discovered by the system. Figure 18 shows the same for Facetednews.

Both Factedpedia and Facetednews were evaluated by the same group of 18 voluntary users. For each system, we partitioned the 12 queries into 3 batches (4 queries in each batch) and asked each

nba	pc game
National_Basketball_Association_draft_picks	Companies_in_the_NASDAQ-100_Index
20th_century_births	Companies_established_in_the_1980s
National_Basketball_Association_players_by_club	Companies_listed_on_the_Hong_Kong_Stock_Exchange
National_Basketball_Association_teams	Dow_Jones_Industrial_Average
United_States_communities_with_African_American	Networking_hardware_companies
Basketball_players_at_the_2004_Summer_Olympics	Companies_established_in_the_1970s
DuPage_County,_Illinois	Companies_based_in_Tokyo
Port_cities_in_the_United_States	Entertainment_Software_Association
Olympic_basketball_players_by_country	20th_century_births
McDonald's_High_School_All-Americans	Companies_listed_on_the_Tokyo_Stock_Exchange
ford	**microsoft**
Car_manufacturers	Companies_in_the_NASDAQ-100_Index
Ford	Liberal_democracies
Liberal_democracies	Living_people
Federal_countries	American_chief_executives
Bus_manufacturers	Circulating_currencies
Truck_manufacturers	Currencies_of_Asia
Companies_based_in_Metro_Detroit	Web_service_providers
G8_nations	Microsoft_employees
Companies_listed_on_the_New_York_Stock_Exchange	Companies_established_in_the_1990s
Motor_vehicle_manufacturers_based_in_Michigan	American_billionaires

Fig. 18: Root categories of 10 facets in the faceted interfaces generated by Facetednews for 4 queries.

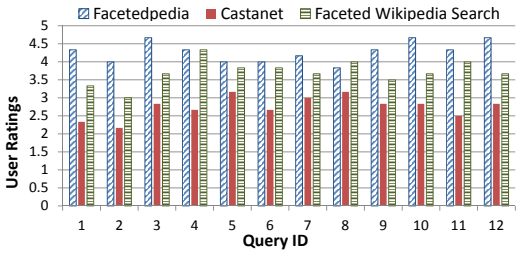
user to participate in one batch in evaluating the system's produced faceted interfaces for the query results. Thus, each batch of queries were evaluated by 6 users.

In evaluating the systems for a query, the query keywords and search objective description were shown to the users. The users were asked to explore the query's target articles using the generated faceted interfaces. For Wikipedia articles, the users were presented three different interfaces generated by Facetedpedia, Castanet, and Faceted Wikipedia Search, respectively. For news articles, the interfaces generated by Facetednews and Castanet were shown. After exploring these interfaces, the users were asked to provide responses in the form of ratings. The ratings were in 5-point scale– 1:"useless", 2: "not that useful", 3:"neutral", 4:"useful to some extent", 5:"very useful". The average ratings over the 12 queries are shown in Figure 19.

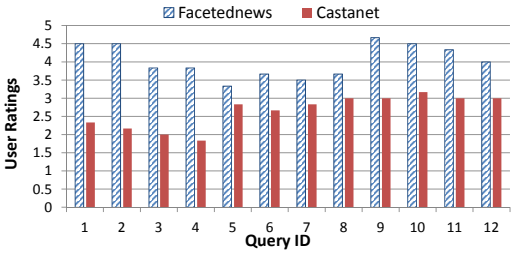
From Figure 19, we see that both Facetedpedia and Facetednews got higher ratings than Castanet in all queries. The results are due to the following main advantages of Facetedpedia over Castanet. First, Facetedpedia uses entities related to documents as their attributes, while Castanet only uses general thesaurus concepts. This makes the facet attributes in Facetedpedia more detailed and specific. Second, Facetedpedia uses Wikipedia category system for hierarchical organization of categories in facets, while to organize the general concepts, Castanet uses WordNet which is not as diverse and rich in semantics as the Wikipedia category system.

Figure 19 also shows that Facetedpedia outperformed Faceted Wikipedia Search in ratings. As discussed in Section 2, Faceted Wikipedia Search uses Wikipedia infoboxes in building facets. The infobox of a Wikipedia article is essentially a collection of attribute-value pairs related to the article. Such structured and table-like metadata makes the generated facets accurate. Given that Facetedpedia directly creates faceted interfaces for text documents without relying on such metadata, the better ratings obtained by Facetedpedia verify the effectiveness of the proposed methods.

In addition to the pre-defined queries, each user was also asked to query the systems with open queries, i.e., arbitrary query keywords that the user came up with during user study. The user was then asked to provide response to three general questions $R1$-$R3$, as follows:

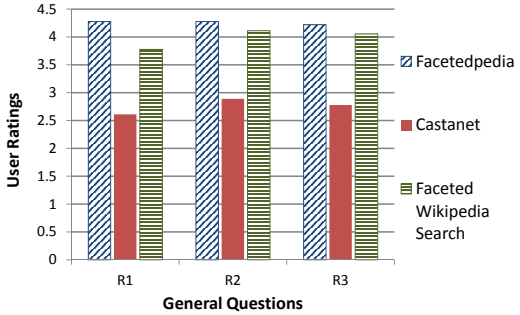


(a) Facetedpedia vs. other systems.

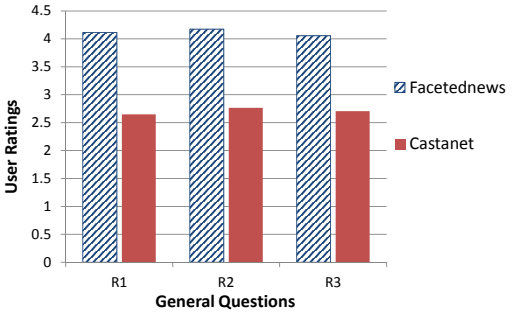


(b) Facetednews vs. other systems.

Fig. 19: Average ratings of compared systems for 12 queries.



(a) Facetedpedia vs. other systems.



(b) Facetednews vs. other systems.

Fig. 20: Average ratings of compared systems for 3 general questions.

— $R1$: Does the system provide diverse and rich facets? (Being "diverse" means the generated facets cover the target articles from different angles and being "rich" means the generated facets have detailed information about the target articles' attributes.)
— $R2$: Does the system provide precise facets? (Being "precise" means the generated facets cover the target articles in a conceptually correct manner.)
— $R3$: Your overall rating of the system.

The responses to these general questions are also ratings at 5-point scale, ranging from 5 (the best score) to 1 (the worst score).

The rating results for the 3 general questions are shown in Figure 20. We see that Facetedpedia and Facetednews received much higher ratings than Castanet on all three questions. With regard to producing diverse and rich interfaces ($R1$), Facetedpedia received stronger ratings than Faceted Wikipedia Search. This is because Faceted Wikipedia Search does not produce query-dependent facets. For target Wikipedia entities in the same domain, it always uses the same set of facets. For instance, if the target entities are people, facets related to *age*, *gender*, *citizenship* will be used, no matter if they are politicians or actors. Hence query-dependent facets such as the movies related to the actors and the political events related to the politicians may not appear as facets. On question $R2$ and $R3$, the ratings of Faceted Wikipedia Search are very close to that of Facetedpedia. This can be due to that Faceted Wikipedia Search creates facets over structured infoboxes. Hence the facets are always accurate to some extent, no matter if the facets are useful for a particular query or not.

	coverage	average pairwise similarity	average category width	average path length
Hill-climbing	**96%**	0.134	**3.4**	**4.5**
Top-k	95%	0.231	3.7	4.8
Random-k	79%	**0.126**	4.5	8.2

Fig. 21: Characteristics of faceted interfaces produced by various algorithms in Facetedpedia.

	coverage	average pairwise similarity	average category width	average path length
Hill-climbing	**93%**	**0.111**	**3.4**	**3.3**
Top-k	91%	0.218	3.5	3.6
Random-k	84%	0.190	31.5	11.7

Fig. 22: Characteristics of faceted interfaces produced by various algorithms in Facetednews.

7.2. Characteristics of Generated Faceted Interfaces

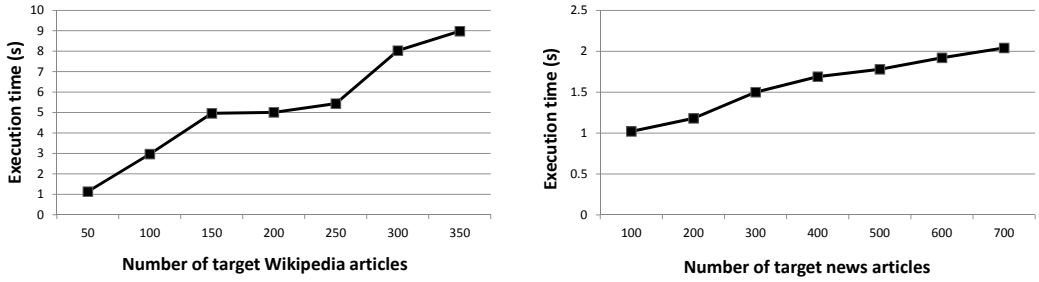
In discovering faceted interfaces, our algorithm in Section 5 optimizes for mainly 3 objectives– high coverage of target articles, low overlap between multiple facets, and low navigational cost. To evaluate if the algorithms meet these objectives, we measured the characteristics of the faceted interfaces produced by our algorithms by four measures, as follows. (1) *Coverage*– the percentage of target articles that can be reached from a faceted interface; (2) *Average pairwise similarity* of the facets in a faceted interface– Equation 10; (3) *Average category width*– the average fan-out of all the categories in all k facets of a faceted interface; (4) *Average path length*– the average length of all possible navigational paths in all k facets of a faceted interface.

We measured these values for three algorithms: *hill-climbing* (Algorithm 4), *top-k* which selects the top k facets ranked by Algorithm 2, and *random-k* which chooses k random facets from the top n facets ranked by Algorithm 2. Figure 21 and Figure 22 show the measured results, for Facetedpedia and Facetednews, respectively. All the values are averaged over the queries listed in Figure 16.

From the results, we see that *hill-climbing* and *top-k* had much better coverage than *random-k*. This verifies that our single-facet ranking (Algorithm 2) is effective in choosing high quality individual facets. In terms of average pari-wise facet similarity, *hill-climbing* and *random-k* performed much better than *top-k*. This verifies that highly ranked facets may substantially overlap. Hence simply choosing the top k facets will result in redundant faceted interfaces, which are worse than even randomly chosen interfaces, in terms of pari-wise similarity. *hill-climbing* achieves not only high coverage but also small overlap. Looking into detailed intermediate results, we observed that the *hill-climbing* method started with choosing top k facets and gradually replaced some facets to make them less similar to each other, while still maintaining the high ranks of chosen facets. In terms of average category width and average path length, *hill-climbing* was slightly better than *top-k* and significantly better than *random-k*, which chose very wide or deep facets from time to time. The average category width and path length attained by *hill-climbing* were around 3 and 4. Therefore the fan-outs of categories and the lengths of navigational paths are within a reasonable range for users.

7.3. Efficiency Evaluation

We evaluated the scalability of our approach by measuring the average execution time of discovering k=20 facets for varying number of target articles (s from 50 to 350 for Facetedpedia and from 100 to 700 for Facetednews). As can be seen from Figure 23, both systems scaled well since the execution time increased linearly with the number of target articles. It also shows that both systems achieved fairly fast response without much performance optimization. (Our facet discovery program was running on a single server without exploiting optimization techniques such as parallel processing for ranking facets and memory caching of data or results.) In average it took 5 seconds for Facetedpedia



(a) Facetedpedia (b) Facetednews

Fig. 23: Execution time of Facetedpedia and Facetednews.

to discover top 20 facets for 200 Wikipedia articles, and 1.5 seconds for Facetednews to discover top 20 facets for 200 news articles. Facetednews ran faster than Facetedpedia, since the attribute entities in news articles are usually not as diverse as the ones in Wikipedia articles. Thus the size of relevant category hierarchy ($\mathcal{RCH}$) in Facetednews is usually much smaller than that in Facetedpedia.

8. RELATED WORK

In Section 2 we compare our work with prior systems that focus on constructing faceted interfaces. In this section we provide further discussion on other aspects of faceted search systems such as personalization, query log, and user behavior modeling. We also provide a brief discussion of related works on querying and exploring Wikipedia.

Koren et al. [Koren et al. 2008] studied how to incorporate user preferences into faceted search by a probabilistic user relevance model. This work also studied how to jump start the personalization by a collaborative user relevance model, when there is no cumulated preferences data for an individual user. In [van Zwol et al. 2010] the authors study faceted exploration of image search results. For building facets, they use the internal semi-structured data sources in a search engine related to images instead of image metadata. For several pre-defined domains (e.g. *locations*, *movies*, etc), they extract a number of relationships (e.g. *subsumes*, *played in*, *has cast*) to form facet dimensions. The work ranks facets based on statistical analysis of image search query logs and users' tagging behavior. Pound et al. [Pound et al. 2011] proposed a query-log mining approach that discovers facets for structured data sources from keyword search query logs. Kules et al. [Kules et al. 2009] applied techniques such as eye tracking, stimulated recall interviews, and direct observation to study user navigation behaviors over faceted interfaces. The results showed that faceted interfaces are useful in searching library catalogs. The study also measured the amount of time that users spend on each component of a faceted interface.

Various approaches have been pursued for enhancing keyword search on Wikipedia. PowerSet [17] uses natural language processing techniques to support simple questions and direct answers. CompleteSearch proactively supports query formulation (by presenting relevant completions) and query refinement through categories (by presenting matching categories) [Bast and Weber 2007]. Several works explicitly support structured queries on Wikipedia. DBPedia [Auer et al. 2007] allows users to ask expressive queries against structured information extracted from Wikipedia. [Chu et al. 2007] uses relational tables to support SQL-style queries over the extracted information. [Zaragoza et al. 2007; Vercoustre et al. 2008] studied how to rank resulting entities of keyword queries. Li et al. [Li et al. 2012] propose a structured query mechanism, entity-relationship query, for searching entities in Wikipedia corpus by their properties and inter-relationships. An entity-relationship query consists of multiple predicates on desired entities. The semantics of each predicate is specified

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

with keywords. Entity-relationship query searches entities directly over text instead of pre-extracted structured data stores. YAGO [Suchanek et al. 2007] supports semantic queries over a knowledge base on Wikipedia. Semantic Wikipedia [Völkel et al. 2006] extends Wikipedia to allow users to manually specify the types of hyperlinks and data values in articles. [Wu and Weld 2007] automatically creates and enhances various structures in Wikipedia, including infoboxes and link structures. Such manually or automatically generated information could be useful in creating faceted interfaces since they explicitly provide the attributes of articles and the relationships between articles.

9. CONCLUSION

The objective of this paper is to develop methods that discover query-dependent faceted interfaces dynamically and automatically to help users navigate keyword search result articles. Toward this goal, we proposed a novel generic faceted interface model that can be instantiated for various application domains. Particularly, we instantiated the generic model into two faceted search systems– Facetedpedia and Facetednews. Both systems use the category system and entities in Wikipedia for building category hierarchies and attribute values in facets. Given the sheer size and complexity of the exploited Wikipedia data, there is a prohibitively 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 over both Wikipedia and news articles.

ACKNOWLEDGMENTS

We thank Lekhendro Lisham and Rakesh Ramegowda for implementing code pieces in Facetedpedia. We thank Dr. Marti Hearst and her group for providing Castanet source code which is used in our experiments. We also thank those who participated in our user studies.

REFERENCES

AUER, S., BIZER, C., KOBILAROV, G., LEHMANN, J., CYGANIAK, R., AND IVES, Z. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference (ISWC'07/ASWC'07)*. 722–735.

BAST, H. AND WEBER, I. 2007. The CompleteSearch engine: Interactive, efficient, and towards IR & DB integration. In *Conference on Innovative Data Systems Research (CIDR)*. Asilomar, CA, USA, 88–95.

BEN-YITZHAK, O., GOLBANDI, N., HAR'EL, N., LEMPEL, R., NEUMANN, A., OFEK-KOIFMAN, S., SHEINWALD, D., SHEKITA, E., SZNAJDER, B., AND YOGEV, S. 2008. Beyond basic faceted search. In *Proceedings of the international conference on Web search and web data mining (WSDM)*. 33–44.

CHU, E., BAID, A., CHEN, T., DOAN, A., AND NAUGHTON, J. 2007. A relational approach to incrementally extracting and querying structure in unstructured data. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, Vienna, Austria, 1045–1056.

CUTTING, D. R., KARGER, D. R., PEDERSEN, J. O., AND TUKEY, J. W. 1992. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*. 318–329.

DAKKA, W. AND IPEIROTIS, P. 2008. Automatic extraction of useful facet hierarchies from text databases. *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE)*, 466–475.

DAKKA, W., IPEIROTIS, P. G., AND WOOD, K. R. 2005. Automatic construction of multifaceted browsing interfaces. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM)*. 768–775.

DEBABRATA, D., JUN, R., MEGIDDO, N., AILAMAKI, A., AND LOHMAN, G. 2008. Dynamic faceted search for discovery-driven analysis. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM)*. 3–12.

DIEDERICH, J. AND BALKE, W.-T. 2008. FacetedDBLP - navigational access for digital libraries. *Bulletin of IEEE Technical Committee on Digital Libraries 4*.

FELLBAUM, C. 1998. WordNet: An electronic lexical database. MIT Press.

HAHN, R., BIZER, C., SAHNWALDT, C., HERTA, C., ROBINSON, S., BÜRGLE, M., DÜWIGER, H., AND SCHEEL, U. 2010. Faceted wikipedia search. In *Business Information Systems*. Springer, 1–11.

HEARST, M. A. 2006. Clustering versus faceted categories for information exploration. *Commun. ACM 49*, 59–61.

KÄKI, M. 2005. Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*. 131–140.

KASHYAP, A., HRISTIDIS, V., AND PETROPOULOS, M. 2010. Facetor: cost-driven exploration of faceted query results. In *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM)*. 719–728.

KOREN, J., ZHANG, Y., AND LIU, X. 2008. Personalized interactive faceted search. In *Proceeding of the 17th international conference on World Wide Web (WWW)*. 477–486.

KULES, B., CAPRA, R., BANTA, M., AND SIERRA, T. 2009. What do exploratory searchers look at in a faceted search interface? In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries (JCDL)*. 313–322.

LI, C., YAN, N., ROY, S. B., LISHAM, L., AND DAS, G. 2010. Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proceedings of the 19th international conference on World wide web (WWW)*. 651–660.

LI, X., LI, C., AND YU, C. 2012. Entity-relationship queries over Wikipedia. *ACM Transactions on Intelligent Systems and Technology (TIST) (In press)*.

MILNE, D. AND WITTEN, I. H. 2008. Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM)*. 509–518.

POLLITT, A. S. 1997. The key role of classification and indexing in view-based searching. In *Proceedings of the 63rd International Federation of Library Associations and Institutions General Conference (IFLA)*.

POUND, J., PAPARIZOS, S., AND TSAPARAS, P. 2011. Facet discovery for structured web search: a query-log mining approach. In *Proceedings of the 2011 international conference on Management of data (SIGMOD)*. 169–180.

PRATT, W., HEARST, M. A., AND FAGAN, L. M. 1999. A knowledge-based approach to organizing retrieved documents. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence (AAAI/IAAI)*. 80–85.

RIJSBERGEN, C. J. V. 1979. *Information Retrieval*, 2nd ed. Butterworth-Heinemann, Newton, MA, USA.

RODDEN, K., BASALAJ, W., SINCLAIR, D., AND WOOD, K. 2001. Does organisation by similarity assist image browsing? In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*. 190–197.

ROSS, K. A. AND JANEVSKI, A. 2005. Querying faceted databases. In *Semantic Web and Databases*, C. Bussler, V. Tannen, and I. Fundulaki, Eds. Lecture Notes in Computer Science, vol. 3372. Springer Berlin / Heidelberg, 199–218.

ROY, S. B., WANG, H., DAS, G., NAMBIAR, U., AND MOHANIA, M. 2008. Minimum effort driven dynamic faceted search in structured databases. In *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM)*. 13–22.

STOICA, E., HEARST, M. A., AND RICHARDSON, M. 2007. Automating creation of hierarchical faceted metadata structures. In *Proceedings of the Human Language Technology Conference (NAACL-HLT)*. 244–251.

SUCHANEK, F. M., KASNECI, G., AND WEIKUM, G. 2007. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web (WWW)*. 697–706.

VAN ZWOL, R., SIGURBJORNSSON, B., ADAPALA, R., GARCIA PUEYO, L., KATIYAR, A., KURAPATI, K., MURALIDHARAN, M., MUTHU, S., MURDOCK, V., NG, P., RAMANI, A., SAHAI, A., SATHISH, S. T., VASUDEV, H., AND VUYYURU, U. 2010. Faceted exploration of image search results. In *Proceedings of the 19th international conference on World wide web (WWW)*. 961–970.

VERCOUSTRE, A.-M., THOM, J. A., AND PEHCEVSKI, J. 2008. Entity ranking in Wikipedia. In *Proceedings of the 2008 ACM symposium on Applied computing (SAC)*. 1101–1106.

VÖLKEL, M., KRÖTZSCH, M., VRANDECIC, D., HALLER, H., AND STUDER, R. 2006. Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web (WWW)*. 585–594.

WU, F. AND WELD, D. S. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM)*. 41–50.

YAN, N., LI, C., ROY, S. B., RAMEGOWDA, R., AND DAS, G. 2010. Facetedpedia: enabling query-dependent faceted search for wikipedia. In *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM)*. 1927–1928.

YEE, K.-P., SWEARINGEN, K., LI, K., AND HEARST, M. 2003. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*. 401–408.

ZAMIR, O. AND ETZIONI, O. 1999. Grouper: a dynamic clustering interface to web search results. In *Proceedings of the eighth international conference on World Wide Web (WWW)*. 1361–1374.

ZARAGOZA, H., RODE, H., MIKA, P., ATSERIAS, J., CIARAMITA, M., AND ATTARDI, G. 2007. Ranking very many typed entities on Wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM)*. 1015–1018.