Extending Faceted Search to the General Web

Weize Kong and James Allan
Center for Intelligent Information Retrieval
School of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
{wkong, allan}@cs.umass.edu

ABSTRACT

Faceted search helps users by offering drill-down options as a complement to the keyword input box, and it has been used successfully for many vertical applications, including e-commerce and digital libraries. However, this idea is not well explored for general web search, even though it holds great potential for assisting multi-faceted queries and exploratory search. In this paper, we explore this potential by extending faceted search into the open-domain web setting, which we call Faceted Web Search. To tackle the heterogeneous nature of the web, we propose to use query-dependent automatic facet generation, which generates facets for a query instead of the entire corpus. To incorporate user feedback on these query facets into document ranking, we investigate both Boolean filtering and soft ranking models. We evaluate Faceted Web Search systems by their utility in assisting users to clarify search intent and find subtopic information. We describe how to build reusable test collections for such tasks, and propose an evaluation method that considers both gain and cost for users. Our experiments testify to the potential of Faceted Web Search, and show Boolean filtering feedback models, which are widely used in conventional faceted search, are less effective than soft ranking models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*query formulation, search process*

General Terms

Algorithms, Experimentation

Keywords

Faceted Web Search, Query Facets, Interactive Feedback

CIKM'14, November 3–7, 2014, Shanghai, China.

ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2661964.

1. INTRODUCTION

Faceted search enables users to navigate a multi-faceted information space by combining text search with drill-down options in each facet. For example, when searching "computer monitor" in an e-commerce site, users can select brands and monitor types from the the provided facets: {*Samsung*, *Dell*, *Acer*, ...} and {*LET-Lit*, *LCD*, *OLED*}. This technique has been used successfully for many vertical applications, including e-commerce and digital libraries.

However, faceted search has not been explored much for general web search, even though it holds great potential for assisting multi-faceted queries and exploratory search [25]. The challenges stem from the large and heterogeneous nature of the web, which makes it difficult to generate and recommend facets [48]. Some recent work [16, 25] extracts facets for a query from the top-ranked search results, providing what appears to be a promising direction for solving the problem. Changing from a global model that generates facets in advance for an entire corpus [46, 13] to a query-based approach that generates facets from the top-ranked documents, these methods not only make the generation problem easier, but also address the facet recommendation problem at the same time. However, their evaluation is based on the similarity between system generated and human created facets, which may not exactly reflect the utility in assisting users' search tasks. Previous work [53] also studied how to use user-selected terms inside the facets for document filtering or re-ranking. However, that study is based on corpora with human-created facet metadata, which is difficult to obtain for the general web.

In this paper, we extend faceted search into the open-domain web setting, which we call Faceted Web Search (FWS). Similar to faceted search, FWS system will provide facets when a user issues a web search query. The user can then select some terms from the facets, which will be used by the FWS system to adjust the search results to better address the user's information need. For example, suppose a user is preparing for an international flight and wants to find baggage allowance information. When the user searches "baggage allowance" in an FWS system, the system may provide a facet for different airlines, {*Delta*, *JetBlue*, *AA*, ...}, a facet for different flight types, {*domestic*, *international*}, and a facet for different classes, {*first*, *business*, *economy*}. When the user selects terms such as "Delta", "international" and "economy" in these facets, the system can ideally help to bring web documents that provide baggage allowance information for the economy class of Delta international flights to the top of the search results.

We describe a way to build an FWS system. To tackle the heterogeneous nature of the web, we propose using query-dependent automatic facet generation which generates facets for a query instead of in advance for the entire corpus. To incorporate user feedback on these query facets into document ranking, we investigate both Boolean filtering and soft ranking models. We also propose an evaluation method for FWS that directly measures the utility in assisting users to clarify search intent and find subtopic information. The evaluation method considers both gain and cost for users. We describe how to build reusable test collections for such tasks, and make our collected data set publicly available. Our experiments show FWS is able to assist the search task and significantly improve ranking performance. Comparing our evaluation with previous evaluations on different facet generation models, we find previous evaluations do not always reflect system utility in real application. Comparing different feedback models, we find Boolean filtering models, which are widely used in conventional faceted search, are too strict in FWS, and less effective than soft ranking models.

The rest of the paper is organized as follows. First, Section2 describes related work. After that, Section 3 describes how an FWS system can be built and introduces the two major components in FWS – query facet generation and facet feedback, which are then described in detail in Sections 4 and 5 respectively. Section 6 describes evaluation for FWS, including previous evaluation approaches and our proposed evaluation. Section 7 presents the experiments.

2. RELATED WORK

2.1 Faceted Search

Previous work on faceted search has studied automatic facet generation [13, 32, 46, 35, 24, 29] and facets recommendation for a query [15, 26]. Most of the work is based on existing facet metadata or taxonomies, and extending faceted search to the general web is still an unsolved problem. The challenges stem from the large and heterogeneous nature of the web [48]. Different from previous work which generates facets for a entire corpus [46, 13], some recent work [16, 25] extracts facets for only a query.

Most evaluations for facet generation/recommendation are either based on comparison between system generated and human created facets [13, 16, 25] or user studies [15, 32, 46]. However, the former may not exactly reflect the utility in assisting users' search tasks, and the latter is expensive to extend for evaluating new systems. In a similar spirit to ours, some work [43, 53, 26] also evaluates facets by their utility in re-ranking documents for users. The differences are their evaluation methods do not capture the time cost for users as explicitly as we do, and their experiments are based on corpora with human created facet metadata. Other evaluations [5, 17, 18, 19, 28] for faceted search are mostly done from a user interface perspective, which is beyond the scope of this paper.

2.2 Query Subtopic/Aspect Mining

Extracting query subtopics (or aspects) is similar to generating facets for queries. A query subtopic is often defined as a distinct information need relevant to the original query. It can be represented as a set of terms that together describe the distinct information need [49, 50, 14] or as a single keyword that succinctly describes the topic [45]. Query subtopics and facets are different in that the terms in a query subtopic are not restricted to be coordinate terms, or have peer relationships, while facets organize terms by grouping "sibling" terms together. For example, {*news*, *cnn*, *latest news*, *mars curiosity news*} is a valid query subtopic for the query *mars landing*, which describes the search intent of Mars landing news, but it is not a valid facet, since the terms in it are not coordinate terms. A valid facet that describes Mars landing news could be {*cnn*, *abc*, *fox*}, which includes different news channels.

2.3 Semantic Class Extraction

Semantic class extraction is to automatically mine semantic classes represented as their class instances from certain data corpus. For example, it may extract *USA*, *UK*, *China* as class instances of semantic class *country*. Due to the similar semantic relationships between terms inside a facet and a semantic class, semantic class extraction can be used for facet generation [25]. The approaches could be roughly divided into two categories: distributional similarity [37, 38, 2, 36] and pattern-based [44].

2.4 Search Results Diversification

Search result diversification has been studied as a method of tackling ambiguous or multi-faceted queries while a ranked list of documents remains the primary output feature of Web search engine today[40]. It tries to diversify the ranked list to account for different search intents or query subtopics. A weakness of search result diversification is that the query subtopics are hidden from the user, leaving him or her to guess at how the results are organized. FWS addresses this problem by providing facets for users to select, and using the explicit feedback to better addressing users' information need.

2.5 Search Results Clustering/Organization

Search results clustering is a technique that tries to organize search results by grouping them into, usually labeled, clusters by query subtopics [6]. It offers a complementary view to the flat ranked list of search results. Most previous work exploited different textual features extracted from the input texts and applied different clustering algorithms with them. Instead of organizing search results in groups, there is also some work [30, 31, 34] that summarizes search results or a collection of documents in a topic hierarchy. For example, Lawrie et al. [30, 31] used a probabilistic model for creating topical hierarchies, in which a graph is constructed based on conditional probabilities of words, and the topic words are found by approximately maximizing the predictive power and coverage of the vocabulary. FWS is different from these work in that it provides facets of a query, instead of directly organizing the search results.

2.6 Facet and Other User Feedback

There is a long history of using user explicit feedback to improve retrieval performance. In relevance feedback [39, 41], documents are presented to users for judgment, after which terms are extracted from the judged relevant document, and added into the retrieval model. In the case where true relevance judgment is unavailable, top documents are assumed to be relevant, which is called pseudo relevance feedback [4, 1]. Because document is a large text unit, which can be difficult for users to judge and for the system to in-

corporate relevance information, previous work also studied user feedback on passages [3, 51] and terms [23, 47]. For faceted search, Zhang et al. [53] study user feedback on facets, using both boolean filtering and soft ranking models. However, the study is based on corpora with human created facet metadata, which is difficult to obtain for the general web. One other difference between our work and most other user feedback work is, facet feedback in our work is used to improve ranking with respect to the query subtopic specified by the feedback terms, instead of the query topic represented by the original query. This presents the scenario in FWS, where users start with a less-specified query, and then use facets to help clarify and search for subtopic information.

3. FACETED WEB SEARCH

We define Faceted Web Search (FWS) as faceted search in the general open-domain web setting. As in ecommerce's faceted search, FWS works by providing facets for users to select among, and adjusting the original ranking according to the user's selection. There are two major components in a FWS system, query facet generation and facet feedback.

Facet generation is typically performed in advance for an entire corpus [46, 13], an approach which is challenging when extended to the general web. Therefore, we use **query facet generation** to generate facets for a query. Using our previous example, for the query "baggage allowance", the system might generate facets, {*Delta*, *JetBlue*, *AA*, ...}, {*domestic*, *international*} and {*first*, *business*, *economy*}.

To differentiate our work with facets in non-web faceted search, we call facets in FWS **query facets**. As shown here, a query facet is a set of coordinate terms – i.e., terms that share a semantic relationship by being grouped under a more general hypernym ("is a" relationship). We call the terms inside facets **facet terms**, which can be single words or phrases. (When it is clear from context, we will simply use "facet" for "query facet", and "term" for "facet term" for convenience.) By changing from generating facets for a global corpus to generating facets only in response to a query, query facet generation not only makes the generation problem easier, but also addresses the facet recommendation problem at the same time. We use and study a few existing query facet generation methods [25, 16], which will be described in detail in Section 4.

Facet feedback is using the facet terms selected by users to adjust the search results. For example, if a user selects the term "Delta", "international" and "economy" in the previous example, the system will then use these terms in the query, ideally to bring web documents that provide baggage allowance information for economy class of Delta international flight to the top of the ranked list. These facet terms selected by users will be called **feedback terms**. Similar to previous work [53], we investigate both Boolean filtering and soft ranking models for facet feedback, which will be described in detail in Section 5

4. QUERY FACET GENERATION

In this section, we describe the query facet generation methods [25, 16] used in this paper. These methods all use top-ranked search results to extract query facets. They work by first extracting candidates from the search results based on predefined patterns and then refining the candidates using different clustering models.

4.1 Extracting Candidates

To extract candidates for query facets, we applied both textual and HTML patterns on the top search results. The patterns used are summarized in Table 1. In the table, all *items* in each pattern are extracted as a candidate list. For example, from the sentence "*... Mars rovers such as Curiosity, Opportunity and Spirit*", according to the lexical pattern, we will extract the candidate facets {*Curiosity*, *Opportunity*, *Spirit*}. For the lexical pattern, we also restrict those *items* to be siblings in the parse tree of that sentence. We use the PCFG parser [22] implemented in Stanford CoreNLP[1] for parsing documents.

Table 1: Facet candidate extraction patterns

Type	Pattern
Lexical	*item*, {,*item*}*, (and|or) {other} *item*
HTML	<select><option>*item*</option>...</select>
	item ...
	item ...
	<table><tr><td>*item*<td>...</table>

After extracting candidate query facets, we further clean the data as follows. First, all the facet terms in the candidates are normalized by converting text to lowercase and removing non-alphanumeric characters. Then we remove stopwords and duplicate terms in each candidate facet. Finally, we remove all candidate facets that contain only one item or more than 200 items.

4.2 Refining Candidates

The candidate query facets extracted are usually noisy [52], and could be non-relevant to the issued query, therefore they need to be refined. Table 2 shows four candidate facets extracted for the query *mars landing*. C_1 contains terms that are relevant to *mars landing*, but they are not coordinate terms: *mars* and *nasa* are not members of the same class. C_2 is a valid query facet, but it is incomplete – another Mars rover *opportunity* appears in C_3. C_3 is extracted from the sentence, "*It is bigger than the 400-pound Mars Exploration rovers, Spirit and Opportunity, which landed in 2004*". As we can see, the term "*the 400 pound mars exploration rovers*" is an extraction error.

Table 2: Four candidate facets for the query *mars landing*

C_1: curiosity rover, mars, nasa, space
C_2: curiosity, opportunity
C_3: the 400 pound mars exploration rovers, spirit, opportunity
C_4: politics, religion, science technology, sports, ...

The methods we used to refine candidate facets basically try to re-cluster the query facets or their facet terms into higher quality query facets. The first type of method is **topic modeling** [52]. We apply both pLSA and LDA on the candidate query facets extracted for a query. The assumption is that, like documents in the conventional setting, candidate facets are generated by a mixture of hidden topics, which are the query facets in our case. After training, the topics are returned as query facets, by using top terms

[1]http://nlp.stanford.edu/software/corenlp.shtml

in each topic. The topic model methods in facet refining only use term co-occurrence information.

The second method is **QDMiner/QDM** [16], which is also an unsupervised clustering method. The method applies a variation of the Quality Threshold clustering algorithm [20] to cluster the candidate facets with bias towards important ones. Then it ranks/selects the facet clusters and the terms in those clusters based on TF/IDF-like scores. This method incorporates more information than just term co-occurrence, but it is not easy to add new features into the model to further improve the performance.

The last group of methods, **QF-I and QF-J**, are supervised methods based on a graphical model, proposed in our previous work [25]. The graphical model learns how likely it is that a term in the candidate facets should be selected in the query facets, and how likely two terms are to be grouped together into a same query facet, using a rich set of features. Then, based on the likelihood scores, QF-I selects the terms and clusters the selected terms into query facets, while QF-J repeats the procedure, trying to performance joint inference. The two methods were shown to be more effective than the other methods, because they incorporate more information into the models and learn from available human labels.

5. FACET FEEDBACK

In this section we describe the two types of models we explore for facet feedback: Boolean filtering and soft ranking. We use t^u to denote a **feedback term** selected by a user u, $F^u = \{t^u\}$ to denote a facet that contains feedback terms (a **feedback facet**), and $\mathcal{F}^u = \{F^u\}$ to denote the set of feedback facets. Given those, a feedback model can be formally denoted as $S'(D, Q, \mathcal{F}^u)$, which gives a score for document D according to the original query Q and the user's feedback $\mathcal{F}^u$.

5.1 Boolean Filtering Model

The Boolean model filters documents based on Boolean operations using the user's feedback $\mathcal{F}^u$. Similar to Zhang et al. [53], we study three different Boolean conditions for filtering. We use the AND condition to require that the document contains *all* of the feedback terms in $\mathcal{F}^u$. The AND condition might be too strict, so a relaxed alternative is to use the OR condition, which requires that the document contains *at least one* of the feedback terms. The last Boolean condition, A+O, is somewhere in between the two conditions above. It use AND across different feedback facets in $\mathcal{F}^u$, and OR for terms t^u inside each facet F^u. The Boolean feedback model scores a document by

$$S'_B(D, Q, \mathcal{F}^u) = \begin{cases} S(D, Q) & \text{if } D \text{ satisfies condition } B(\mathcal{F}^u) \\ -\infty & \text{otherwise} \end{cases} \quad (1)$$

where condition B can be either AND, OR, or A+O, and $S(D, Q)$ is the score returned by the original retrieval model. Notice that when there is only a single feedback term, the three conditions will be equivalent; when there is only one feedback query facet (group of feedback terms), AND and A+O will be equivalent.

5.2 Soft Ranking Model

While Boolean filter models are commonly used in faceted search, it may be too strict for FWS. The Boolean filtering model is based on two assumptions [53]: (1) users are clear about what they are looking for, and thus are able to select proper feedback terms to restrict the results; and (2) matching between a facet term and a document is accurate and complete. In FWS, that means a document that contains the feedback term should be relevant to the term, and all documents relevant to that feedback term should contain the term. However, both of the two assumptions are unlikely to hold in FWS.

For that reason, we also use soft ranking models, which expand the original query with feedback terms, using a linear combination as follows

$$S'_E(D, Q, \mathcal{F}^u) = \lambda S(D, Q) + (1 - \lambda) S_E(D, \mathcal{F}^u) \quad (2)$$

where $S(D, Q)$ is the score from the original retrieval model as before, $S_E(D, \mathcal{F}^u)$ is the expansion part which captures the relevance between the document D and feedback facet $\mathcal{F}^u$, using expansion model E. λ is a parameter for adjusting the weight between the two parts.

We use two expansion models, a term and a facet expansion model. The term expansion model, ST, assigns equal weight for all the feedback terms, as follow,

$$S_{ST}(D, \mathcal{F}^u) = \frac{1}{N} \sum_{F^u \in \mathcal{F}^u} \sum_{t^u \in F^u} S(D, t^u) \quad (3)$$

where N is the total number of facet terms. $S(D, t^u)$ can be the original retrieval model used for the query or a different model.

The facet expansion model, SF, uses the facet structure information. It assigns equal weights between each feedback facets, and equal weights between feedback terms within the same facet, as shown below.

$$S_{SF}(D, \mathcal{F}^u) = \frac{1}{|\mathcal{F}^u|} \sum_{F^u \in \mathcal{F}^u} \frac{1}{|F^u|} \sum_{t^u \in F^u} S(D, t^u) \quad (4)$$

Notice that the two expansion models will be equivalent when there is only a single feedback term or when there is only one feedback facet. In our experiments, we use the Sequential Dependence Model (SDM) [33] as the baseline retrieval model $S(D, Q)$, which incorporates word unigrams, adjacent word bigrams, and adjacent word proximity. For $S(D, t^u)$, we use the Query Likelihood model with Dirichlet smoothing as below,

$$S(D, t^u) = \sum_{w \in t^u} \log \frac{tf(w, D) + \mu \frac{tf(w, \mathcal{C})}{|\mathcal{C}|}}{|D| + \mu} \quad (5)$$

where w is a word in t^u, $tf(w, D)$ and $tf(w, \mathcal{C})$ are the number of occurrences of w in the document and the collection respectively; μ is the Dirichlet smoothing parameter; $|D|$ is the number of word in $|D|$, and $|\mathcal{C}|$ is the total number of words in the collection.

6. EVALUATION

Previous work [16, 25] evaluated query facet generation by comparing generated facets with human annotated ones on the theory that mimicking a person on selection is the right choice. However, this intrinsic evaluation does not necessarily reflect the *utility* of the generated facets in assisting search – that is, some annotator-selected facets may be of little value for the search task, and some good facet terms may be missed by annotators. In an effort to address those issues, we propose here an extrinsic evaluation method to directly measures the utility based on a FWS task.

6.1 Intrinsic Evaluation

In the intrinsic evaluation, "gold standard" query facets are constructed by human annotators and used as the ground truth to be compared with facets generated by different systems. The facet annotation is usually done by first pooling facets generated by the different systems. Then annotators are asked to group or re-group terms in the pool into preferred query facets, and to give ratings for each of them regarding how useful or important the facet is.

Conventional clustering metrics, such as Purity and Normalized Mutual Information/NMI, are used in intrinsic evaluation, as well as newly designed metrics for facet generation, including $wPRF_{\alpha,\beta}$ [25] and some variations of nDCG [16]. $wPRF_{\alpha,\beta}$ combines wP, wR and wFP, which are weighted version of precision and recall for facet terms, and the F1 measure for facet term clustering. Parameters α and β are used to adjust the emphasis of the three factors, but simply set to one in the reported experiments. However, that metric does not account for facet ranking performance. In the nDCG variation metrics, system facets are mapped to truth facets, and assigned ratings according to their mapped truth facets. Then the ranked system facets are evaluated using nDCG, with the discounted gain further weighted by the precision and recall of the system facet and mapped truth facet. In this work, we slightly alter the weighting by using an F1 measure to combine precision and recall together instead of multiplying them together as in the original work [16]. We call this variant metric f1-nDCG.

6.2 Extrinsic Evaluation

The intrinsic evaluation is not based on any particular search task, and thus may not reflect the real utility of the generated facets in assisting search. Therefore, we propose an extrinsic evaluation method which evaluates a system based on an interactive search task that incorporates FWS. We believe the task is similar to a real application of FWS: a user searches using an under-specified query, the FWS system provides query facets from which the user can select feedback terms that would help further specified the query, after which the FWS system uses the feedback terms for re-ranking documents.

For the evaluation, ideally we could ask real users or carry out user studies to try each of FWS systems, and measure the gain and cost for using them. The gain can be measured by the improvement of the re-ranked results using standard IR metrics like MAP or nDCG. The cost can be measured by the time spent by the users giving facet feedback. However this evaluation is difficult and expensive to extend for evaluating new systems rapidly.

We instead propose to *simulate* the user feedback process based on an interaction model, using oracle feedback terms and facet terms collected from annotators. Both the oracle feedback and annotator feedback incrementally select all feedback terms that a user may select, which will then be used in simulation based on the user model to determine which subset of the oracle or annotator feedback terms are selected by a user and how much time is spent giving that feedback. Finally, the systems are evaluated by the re-ranking performance together with the estimated time cost.

For the simulated FWS task, we use the TREC Web track dataset of the diversification task [7, 8, 9, 10]. It includes query topics that are structured as a representative set of subtopics, each related to a different user need, with relevance judgment made at the subtopic level. In our task, each subtopic is regarded as the search intent of a user, and the corresponding topic title is used as the under-specified query issued to the FWS system. For example, for the number 10 query in TREC 2009 Web Track, the title "cheap internet" is used as the initial query, and its subtopic "I want to find cheap DSL providers" is regarded as the search intent of the user.

6.2.1 Oracle and Annotator Feedback

Oracle feedback presents an ideal case of facet feedback, in which only *effective* terms – those that improve the quality of the ranked list – are selected as feedback terms. We extract oracle feedback terms by testing each single term in the presented facets. Each single candidate term is used by a facet feedback model to re-rank the documents and the candidate term is selected for the oracle if the improvement of the re-ranked documents meets a threshold. In our experiment, we use MAP as the metric and set the threshold to be 0.01. Since we have two types of feedback models, there are two sets of oracle feedback terms – one uses the Boolean filter models, and one uses the soft ranking models.

Oracle feedback is cheap to obtain for any facet system (assuming document relevance judgments are available), however it may be quite different from what actual users may select in a real interaction. Therefore, we also collect feedback terms from annotators. The facet feedback annotation is done by presenting the facets to an annotator with description of the information need and the initial under-specified query. The annotator is asked to select all the terms from the facets that would help address the information need. Ideally, we could present all the facets generated from different FWS systems for this annotation, but it would be quite expensive. In our experiment, we only present annotator facets collected from the intrinsic evaluation. This assumes all other facet terms generated by systems are uninteresting to the user or at least not easy for the user to select.

6.2.2 User Model

The user model describes how a user selects feedback terms from facets, based on which we can estimate the time cost for the user. While any reasonable user model can be brought to play here, we use a simple one, similar to the user model others have used for selecting suggestions from clusters of query auto-completions [21].

Our user model is based on the structural property of facets. By grouping terms into facets, the facet interface essentially provides a skip list of these facet terms for users. More specifically, in the model, a user sequentially scans presented query facets and skips an entire facet if the user finds the facet irrelevant. Otherwise, the user will scan within the facet, sequentially reading and selecting desired facet terms, until the user finds the desired one or ones. Based on this user model, the time cost for giving facet feedback can be calculated as as,

$$T(\mathcal{F}^u) = \sum_{F^u \in \mathcal{F}^u} \left(T_f(F^u) + \sum_{t \in ts(F^u)} T_t(t) \right) \qquad (6)$$

The righthand side of the equation contains two parts. The first part $T_f(F^u)$ is the time for scanning a facet and deciding relevance, and the second part is the time for scan-

ning/selecting terms in the relevant facets. $ts(F^u)$ is the set of terms scanned/selected in F^u's corresponding query facet, and $T_t(t)$ is the time used for scanning/selecting a term. Since we assume users sequentially scan the terms inside a facet, $ts(F^u)$ will include all the beginning terms in F^u's corresponding facet until the last selected term. This is based on the assumption that users are clear about what terms to select, and stop scanning after finding all of them.

To simplify the estimation, we further assume time costs are equal for scanning different facets, and equal for scanning/selecting different terms. Then the estimation becomes

$$T(\mathcal{F}^u) = |\mathcal{F}^u| \cdot T_f + |ts(\mathcal{F}^u)| \cdot T_t \quad (7)$$

where $|ts(\mathcal{F}^u)| = \sum_{F^u \in \mathcal{F}^u} |ts(F^u)|$ is the total number of term scanned/selected. T_f and T_t are now parameters representing the time for scanning a facet and time for scanning/selecting a term respectively.

To estimate parameters T_f and T_t, we tracked annotator behavior during the feedback annotation described in Section 6.2.1, including selecting / un-selecting terms and starting / exiting an annotation session. We only used annotation sessions which did not contain any un-selecting actions, and filter out some inappropriate sessions, e.g. the annotator dwells for a long time with no activity. This selection results in 274 annotator sessions. We then extracted $|\mathcal{F}^u|$ and $|ts(\mathcal{F}^u)|$ as well as the time cost $T(\mathcal{F}^u)$ for each session, and used linear regression to fit the model to the data. When using sessions from all annotators, T_f and T_t are estimated as 2.60 and 1.60 seconds respectively, with $R^2 = 0.089$. The low R^2 is partly due to the variance introduced by using sessions of different annotators. When using one single annotator we obtain a better fit with $R^2 = 0.555$, and $T_f = 1.51$, $T_t = 0.66$, for one of the annotators. Since the estimation for T_f is about twice of T_t, for simplicity, in our experiment, we set $T_f = 2 \cdot T_t$, and report the time cost in the time unit of reading/scanning a single term.

Based on this user model, given oracle/annotator feedback, which represents all the terms that a user may select, the extrinsic evaluation works as follows. We incrementally include each term in oracle/annotator feedback as a feedback terms, and measure how ranking performance changes together with the time cost estimated based on the user model.

7. EXPERIMENTS

7.1 Experiment Settings

Data set. For the document corpus, we use the ClueWeb09 Category-B collection and apply spam filtering with a threshold of 60 using the Waterloo spam scores [12]. The spam-filtered collection is stemmed using the Krovetz stemmer [27]. For the query topics and subtopics, we used those from TREC Web Track's diversity task from 2009 to 2012, which also contain relevance judgments for documents with respect to each subtopic. We constrain the subtopics to have at least one relevant document in the spam-filtered collection, and this results in 196 queries and 678 query subtopics in our experiment set. For the relevance judgment, any documents that are not in the spam-filtered collection are discarded.

Annotation. We collected facet annotations as described in Section 6.1 for all 196 queries. Facets are pooled from the top 10 facets generated by runs from QDM, pLSA, LDA, QF-I and QF-J. Then annotators are asked to group the terms in the pool into query facets, and to give a rating for the query facet using a scale of good (2) or fair (1). Facet annotation statistics are given in Table 3.

Table 3: Facet annotation statistics

	fair	good	pooled
#terms per query	15.8	26.5	240.0
#facets per query	2.3	3.8	40.9
#terms per facet	6.8	6.9	5.9

For the extrinsic evaluation, we also collected facet feedback annotations as described in Section 6.2 for all 678 subtopics. The statistics are given in Table 4, which also includes statistics for oracle feedback. The table shows the number of feedback terms selected per subtopic and the number of feedback facets per subtopic. For some subtopics, there may be no feedback terms selected, so we also report feedback coverage over subtopics in the table.

Table 4: Oracle and annotator feedback statistics. oracle-b and oracle-s are oracle feedback based on the Boolean filter model and soft ranking model respectively.

	annotator	oracle-b	oracle-s
#fdbk terms/subtopic	4.10	7.83	5.24
#fdbk facet/subtopic	1.36	2.40	1.93
feedback coverage	0.80	0.74	0.72

Training/testing and parameter tuning are based on 4-fold cross validation for the same splits of the 196 queries.

Significance test is performed by using paired t-test, using 0.05 as the p-value threshold.

Facet Generation Models. We compare pLSA, LDA, QDM, QF-I and QF-J. $wPRF$ ($wPRF_{\alpha,\beta}$ with α and β set to 1.0) is used as the metric for parameter tuning. For pLSA and LDA, we tune the number of facets and the number of facet terms in a facet. For QDM we tune the two parameters used in the clustering algorithm, the diameter threshold for a cluster and the weight threshold for a valid cluster, as well as the parameters they used for selecting facet terms in each facet. For QF-I and QF-J, we do not use the features based on snippets (which was used in the previous work [25]), since snippets are not available in our system. For QF-I, we tune the weight threshold for facet terms, and the diameter threshold. For QF-J, there are no parameter that need to be tuned.

Baseline Retrieval Models and Facet Feedback Models. We use SDM as the baseline retrieval model with 0.8, 0.15, 0.05 weights for word unigrams, adjacent word bigrams, and adjacent word proximity respectively. SDM is also used as the initial retrieval model for facet generation and facet feedback. We compare different facet feedback models to SDM, including AND, OR, A+O for the Boolean filtering models, as well as ST and SF for the soft ranking models. λ in ST/SF is set to be 0.8. Dirichlet smoothing $\mu = 1500$ is used for both SDM and ST/SF. We also used other baselines including RM3 [1], a pseudo relevance feedback model, tuned on MAP, and xQuAD [42], a diversification model, tuned on α-NDCG [11].

7.2 Comparison to Baseline Retrieval Models

We first compare FWS with other baseline retrieval models in Table 5. QF-I is used as the FWS system here, with SF as the facet feedback model. Annotator feedback terms are used, which represents a real case (not oracle) of FWS application. In the table, QF-I:10 and QF-I:50 are QF-I runs allowed 10 and 50 time units for feedback respectively.

First, the table shows that using annotator feedback, QF-I can improve ranking over the initial retrieval model, SDM. QF-I also obtains better results than RM3, across all the metrics. It is also better than xQuAD for most metrics. The observations testify to the potential of FWS in assisting search. Last, when allowed more time, the results are further improved as shown by the change from QF-I:10 to QF-I:50.

Table 5: Retrieval effectiveness comparison with baselines. QF-I:10 and QF-I:50 are QF-I runs allowed 10 and 50 time units for feedback respectively. Statistically significant differences are marked using the first letter of the retrieval model name under comparison.

Model	MAP	MRR	nDCG@10
SDM	0.1854	0.3295	0.1997
RM3	0.1886	0.3124	0.2010
xQuAD	0.1822	0.3463^{r}	0.2191
QF-I:10	0.1918_{x}^{s}	$0.3476^{s,r}$	$0.2145^{s,r}$
QF-I:50	$0.2044_{x}^{s,r}$	$0.3736^{s,r}$	$0.2357^{s,r}$

7.3 Oracle and Annotator Feedback

In Figure 1, we compare the effectiveness of oracle and annotator feedback. It shows how ranking performance changes as time cost increases, when incrementally including terms from the two types of feedback as feedback terms. The time cost is estimated by the user model described in Section 6.2.2. MAP is calculated with respect to the subtopic level relevance, since we are evaluating the case where the user is looking for the subtopic information. MAP value is averaged by macro-averaging – averaging for subtopics within the same query first, and then across all the queries. [2] When time is zero, no feedback terms are used, which is then just the result for the initial ranking from SDM.

In Figure 1, MAP increases from the SDM baseline result for both oracle and facet feedback, with the oracle ones shown to be far more effective. This shows that annotators are able to identify some useful feedback terms, but not as effective as the ideal case: it seems people have a hard time knowing which terms are most likely to be successful. We further compare the feedback terms selected in oracle and annotator feedback in Table 6, which also supports this claim.

Table 6 shows the overlap between oracle and annotator feedback is low according to F1. However, annotators are able to find almost half of the oracle feedback terms. Other oracle feedback terms are difficult for annotators (or users) to recognize, due to lack of background knowledge, or underlying statistical dependencies between words that are difficult to capture. For example, for the query subtopic, "find the TIME magazine photo essay Barack Obama's Family Tree", some names of family members are selected in oracle feedback, but not by the annotator. This is because the annotator is not able to capture the relevant relationship between the names of family members and the photo essay, or simply because the annotator does not know those family members' names.

[2]We also measured micro-averaging, but the results are similar.

Figure 1: MAP change over time for oracle and annotator feedback, based on annotator facets and SF feedback model. oracle-s and oracle-b are the oracle feedback based on the Boolean filtering model and soft ranking model respectively.

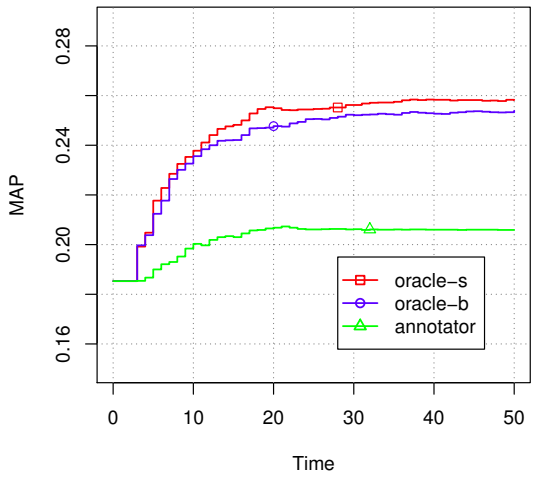


Table 6: Comparing feedback terms in annotator feedback and oracle feedback, using oracle-s as ground truth. This table shows that the annotator selects only 44% of the effective terms and that only 28% of the selected terms are effective.

Precision	Recall	F1
0.2817	0.4412	0.2179

7.4 Comparing Facet Generation Models

7.4.1 *Intrinsic Evaluation*

In Table 7 we evaluate different facet generation models using intrinsic evaluation. The table shows QF-I and QF-J outperform other models on the overall measure, wPRF. QF-I wins because of high recall of facet terms and high F1 of facet term clustering. For f1-nDCG, QF-J and QDM are more effective. These results are consistent with our previous work [25].

Table 7: Intrinsic evaluation of facet generation models.

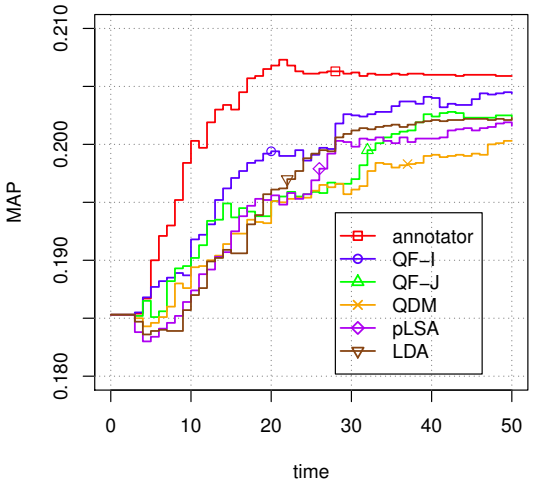
Model	wP	wR	wFP	wPRF	f1-nDCG
pLSA	0.2198	0.6273	0.2541	0.2521	0.1430
LDA	0.2720	0.5578	0.2345	0.2571	0.1267
QDM	0.3253	0.4024	0.2492	0.2688	0.1782
QF-J	0.3525	0.4060	0.2779	0.2836	0.2269
QF-I	0.2729	0.7363	0.3859	0.3448	0.1602

7.4.2 *Extrinsic Evaluation*

Intrinsic evaluation may not reflect the utility of facets in assisting search. In Figure 2 we evaluate different facet generation models using extrinsic evaluation, by showing how MAP changes as time cost increases, similar to Figure 1.

First, Figure 2 shows all models are able to improve ranking from the baseline, which testifies to the potential of FWS. However, the automatically generated facets are less effective than annotator facets. MAP for annotator facets reaches 0.2 by 10 time units, while the models need much more time, ranging from 27 to 47. Second, QF-I is more effective than other models over the entire time span. This is consistent with the intrinsic evaluation. Third, the comparison results for other models is less clear. QF-J and QDM are better than pLSA and LDA before 20 time units, but MAP for pLSA and LDA increases much faster afterwards, and ended at a value similar to QF-J. Comparing these results with Table 7, we find intrinsic metrics do not always reflect utility based on extrinsic evaluation.

Figure 2: MAP change over time for different facets generation models, based on annotator feedback and SF feedback model.



Another way to compare is to see how may terms in the presented facets are selected by annotators, as shown in Figure 3. The figure shows that with annotator facets a (simulated) user needs less time for selecting feedback terms. All the other facet generation approaches are similar to each other, with QDM having slightly more feedback terms at the beginning and QF-I having more for the rest. This explains why QF-I is the best system run in Figure 3 – for the same time cost, QF-I has more feedback terms selected by annotators.

If we switch to using the oracle feedback facets, the difference between different facet generation models and annotator facets are no longer that big, as shown in Figure 4. Annotator facets are better at the beginning, but the corresponding MAP stops growing at around 20 time units. We find this is due to there not being so many facets available in the annotator facets.

The number of terms and facets presented to users will affect this evaluation. In the plot, when there is not a suffi-

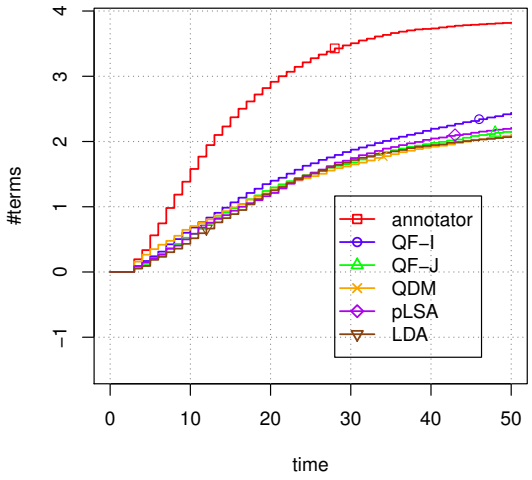


Figure 3: Number of feedback terms selected over time on facets generated by different models, based on annotator feedback.

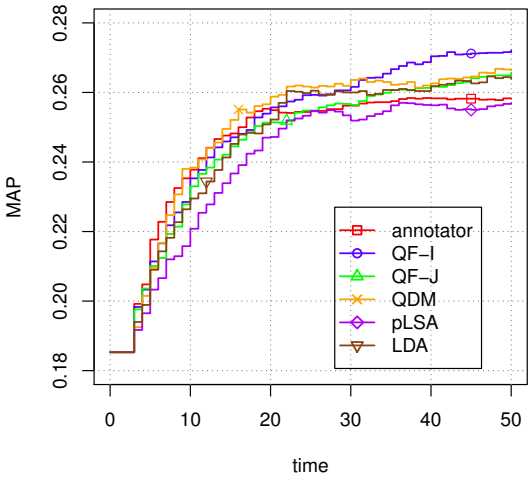
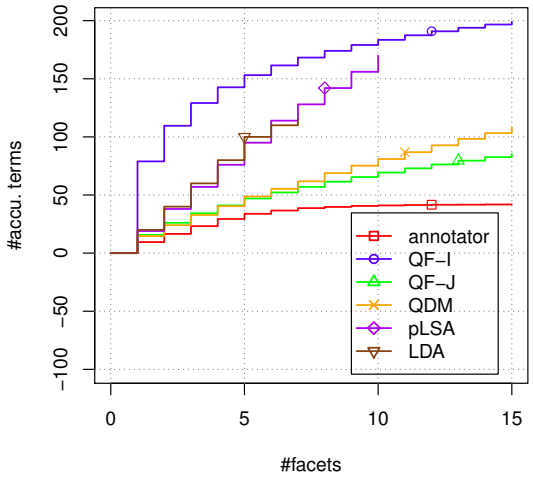


Figure 4: MAP change over time for different facets generation models, based on oracle feedback and SF feedback model.

cient supply of facets at some time cost, the results from a smaller time cost is used. That is, if the user runs out of facet terms to consider, performance is stuck where it last left off. To validate the comparison in Figure 2 and 4, we plot Figure 5 which shows the number of accumulated facet terms in the top facets generated by different models. Figure 5 shows all models have a sufficient supply of facet terms for these evaluations. All of them present at least 50 facet terms (on average), which will need at least 50 time units for the user to process. This obviates the concern above. However, the annotator only has on average 42.3 facet terms selected, and therefore comparison at a time larger than that might unfairly penalize the annotator facets. We also notice that the first facet in QF-I is very large, and overall QF-I has

Figure 5: Accumulated number of facet terms in top facets generated from different facets generation models.



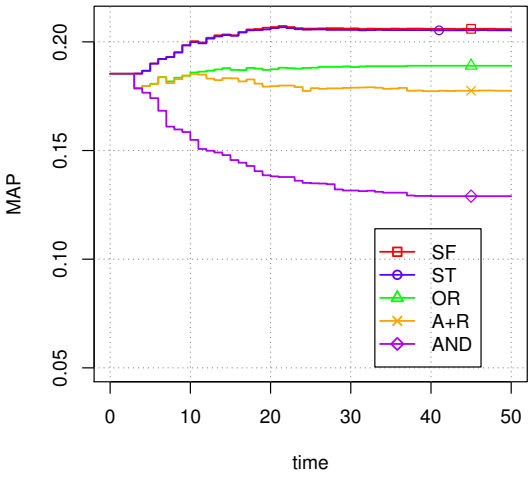


Figure 6: MAP change over time for different feedback models, based on annotator facets and annotator feedback.

more terms in top facets. Since the results are tuned on wPRF with equal weight for term precision and recall, this suggests it is very likely that too much weight is assigned for recall, and a more balanced weight between wP, wR and wFP should be used in wPRF.

7.5 Comparing Facet Feedback Models

We compare different facet feedback models in Figure 6. It shows soft ranking models are more effective than Boolean filter models. AND is too aggressive, which hurts the ranking performance as more and more feedback terms are used. The other two Boolean filtering models, OR and A+R, are similar at the beginning. That is because in the beginning there is only one feedback facet, in which case OR and A+R will be equivalent. As more facet terms are selected, A+R performance decreases. For the two soft ranking model, SF and ST are very close, with SF slightly better as time progresses. This comparison suggests that Boolean filter models, AND and A+O, are too strict for FWS, and a soft ranking model is more effective for FWS. This situation is probably because in FWS the mapping between facet term and document is incomplete; a document that does not contain the exact facet term may also be relevant.

7.6 Examples

In this section, we use some system generated facets as examples, to show how FWS can assist search. We find FWS can be helpful in exploratory search. For example, for the query "cheap internet", the facets generated by QDM includes a facet of different Internet service types, {*dial up*, *dsl*, *cable*}, and a facet of different ISPs, {*netzero*, *juno*, *copper*, *toast*}. These facets can assist the user to compare different Internet service types and ISPs during his/her exploration of "cheap Internet". Another example is the query "lymphoma in dogs", in which the user may want to learn about different aspects of lymphoma in dogs. QF-J generates facet {*treatment*, *diagnosis*, *prognosis*, *symptoms*, ...} which represents different aspects of the query. For this query, there is a query subtopic looking for symptoms of lymphoma in dogs, which can be directly answered by another facet found by QF-J, {*vomiting*, *diarrhea*, *weight loss*, *depression*, *fever*}.

8. CONCLUSIONS

In this paper, we proposed Faceted Web Search, an extension of faceted search to the general Web. We studied different facet generation and facet feedback models based on our proposed extrinsic evaluation, which directly measures the utility in search instead of comparing system/annotator facets as in intrinsic evaluation. We also describe a way to build reusable test collection for the extrinsic evaluation, and make our collected data set publicly available[3].

Our experiments show, by using facet feedback from users, Faceted Web Search is able to assist the search task and significantly improve ranking performance. Comparing intrinsic evaluation and extrinsic evaluation on different facet generation models, we find that the intrinsic evaluation does not always reflect system utility in real application. Comparing different facet feedback models, we find that the Boolean filtering models, which are widely used in conventional faceted search, are too strict in Faceted Web Search, and less effective than soft ranking models.

9. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by IBM subcontract #4913003298 under DARPA prime contract #HR001-12-C-0015. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

[3]See http://ciir.cs.umass.edu/downloads

10. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at trec 2004: Novelty and hard. Technical report, DTIC Document, 2004.

[2] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proc. of NAACL-HLT*, pages 19–27, 2009.

[3] J. Allan. Relevance feedback with too much data. In *Proc. of SIGIR*, pages 337–343, 1995.

[4] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using smart: Trec 3. *NIST special publication*, pages 69–69, 1995.

[5] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *Proc. of National Conference of Artificial Intelligence*, 1996.

[6] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17, 2009.

[7] C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. Technical report, DTIC Document, 2009.

[8] C. L. Clarke, N. Craswell, I. Soboroff, and G. V. Cormack. Overview of the trec 2010 web track. Technical report, DTIC Document, 2009.

[9] C. L. Clarke, N. Craswell, I. Soboroff, and E. M. Voorhees. Overview of the trec 2011 web track. Technical report, DTIC Document, 2009.

[10] C. L. Clarke, N. Craswell, and E. M. Voorhees. Overview of the trec 2012 web track. Technical report, DTIC Document, 2009.

[11] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proc. of SIGIR*, pages 659–666, 2008.

[12] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

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

[14] V. Dang, X. Xue, and W. B. Croft. Inferring query aspects from reformulations using clustering. In *Proc. of CIKM*, pages 2117–2120, 2011.

[15] D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman. Dynamic faceted search for discovery-driven analysis. In *Proc. of CIKM*, pages 3–12, 2008.

[16] Z. Dou, S. Hu, Y. Luo, R. Song, and J.-R. Wen. Finding dimensions for queries. In *Proc. of CIKM*, pages 1311–1320, 2011.

[17] J. English, M. Hearst, R. Sinha, K. Swearingen, and K.-P. Yee. Hierarchical faceted metadata in site search interfaces. In *Proc. of CHI*, pages 628–639, 2002.

[18] M. Hearst. Design recommendations for hierarchical faceted search interfaces. In *SIGIR Workshop on Faceted Search*.

[19] M. Hearst. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In *Workshop on Computer Interaction and Information Retrieval, HCIR*, 2008.

[20] L. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome research*, 9(11):1106–1115, 1999.

[21] A. Jain and G. Mishne. Organizing query completions for web search. In *Proc. of CIKM*, pages 1169–1178, 2010.

[22] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proc. of ACL*, pages 423–430. Association for Computational Linguistics, 2003.

[23] J. Koenemann and N. J. Belkin. A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *Proc. of SIGCHI*, pages 205–212, 1996.

[24] C. Kohlschütter, P.-A. Chirita, and W. Nejdl. Using link analysis to identify aspects in faceted web search. In *SIGIR Workshop on Faceted Search*, 2006.

[25] W. Kong and J. Allan. Extracting query facets from search results. In *Proc. of SIGIR*, pages 93–102, 2013.

[26] J. Koren, Y. Zhang, and X. Liu. Personalized interactive faceted search. In *Proc. of WWW*, pages 477–486, 2008.

[27] R. Krovetz. Viewing morphology as an inference process. In *Proc. of SIGIR*, pages 191–202, 1993.

[28] B. Kules, R. Capra, M. Banta, and T. Sierra. What do exploratory searchers look at in a faceted search interface? In *Proc. of JCDL*, pages 313–322, 2009.

[29] K. Latha, K. R. Veni, and R. Rajaram. Afgf: An automatic facet generation framework for document retrieval. In *Proc. of ACE*, pages 110–114. IEEE, 2010.

[30] D. Lawrie, W. B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *Proc. of SIGIR*, pages 349–357, 2001.

[31] D. J. Lawrie and W. B. Croft. Generating hierarchical summaries for web searches. In *Proc. of SIGIR*, pages 457–458, 2003.

[32] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das. Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proc. of WWW*, pages 651–660, 2010.

[33] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc. of SIGIR*, pages 472–479, 2005.

[34] C. G. Nevill-Manning, I. H. Witten, and G. W. Paynter. Lexically-generated subject hierarchies for browsing large collections. *International Journal on Digital Libraries*, 2(2-3):111–123, 1999.

[35] E. Oren, R. Delbru, and S. Decker. Extending faceted navigation for rdf data. In *Proc. of ISWC*, pages 559–572, 2006.

[36] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *Proc. of EMNLP*, pages 938–947, 2009.

[37] P. Pantel and D. Lin. Discovering word senses from text. In *Proc. of SIGKDD*, pages 613–619, 2002.

[38] P. Pantel, D. Ravichandran, and E. Hovy. Towards terascale knowledge acquisition. In *Proc. of ICCL*, page 771. Association for Computational Linguistics, 2004.

[39] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[40] T. Sakai and R. Song. Evaluating diversified search results using per-intent graded relevance. In *Proc. of SIGIR*, pages 1043–1052, 2011.

[41] G. Salton. Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24:5, 1997.

[42] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proc. of WWW*, pages 881–890, 2010.

[43] A. Schuth and M. Marx. Evaluation methods for rankings of facetvalues for faceted search. In *Multilingual and Multimodal Information Access Evaluation*, pages 131–136. 2011.

[44] S. Shi, H. Zhang, X. Yuan, and J.-R. Wen. Corpus-based semantic class mining: distributional vs. pattern-based approaches. In *Proc. of ICCL*, pages 993–1001, 2010.

[45] R. Song, M. Zhang, T. Sakai, M. Kato, Y. Liu, M. Sugimoto, Q. Wang, and N. Orii. Overview of the ntcir-9 intent task. In *Proc. of NTCIR-9 Workshop Meeting*, pages 82–105, 2011.

[46] E. Stoica and M. A. Hearst. Automating creation of hierarchical faceted metadata structures. In *In Procs. of NAACL-HLT*, 2007.

[47] B. Tan, A. Velivelli, H. Fang, and C. Zhai. Term feedback for information retrieval with language models. In *Proc. of SIGIR*, pages 263–270, 2007.

[48] J. Teevan, S. Dumais, and Z. Gutt. Challenges for supporting faceted search in large, heterogeneous corpora like the web. *Proc. of HCIR*, pages 6–8, 2008.

[49] X. Wang, D. Chakrabarti, and K. Punera. Mining broad latent query aspects from search sessions. In *Proc. of SIGKDD*, pages 867–876, 2009.

[50] F. Wu, J. Madhavan, and A. Y. Halevy. Identifying aspects for web-search queries. *JAIR*, 40:677–700, 2011.

[51] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proc. of SIGIR*, pages 4–11, 1996.

[52] H. Zhang, M. Zhu, S. Shi, and J.-R. Wen. Employing topic models for pattern-based semantic class discovery. In *Proc. of the ACL-IJCNLP*, pages 459–467, 2009.

[53] L. Zhang and Y. Zhang. Interactive retrieval based on faceted feedback. In *Proc. of SIGIR*, pages 363–370, 2010.