

Search Result Presentation Based on Faceted Clustering

Benno Stein

Tim Gollub

Dennis Hoppe

Bauhaus-Universität Weimar
99421 Weimar, Germany
<first>.<last>@uni-weimar.de

ABSTRACT

We propose a competence partitioning strategy for Web search result presentation: the unmodified head of a ranked result list is combined with a clustering of documents from the result list tail. We identify two principles to which such a clustering must adhere to improve the user’s search experience: (1) Avoid the unwanted effect of query aspect repetition, which is called *shadowing* here. (2) Avoid extreme clusterings, i.e., neither the number of cluster labels nor the number of documents per cluster should exceed the size of the result list head. We present measures to quantify the shadowing effect, and with Faceted Clustering we introduce an algorithm that optimizes the identified principles. The key idea of Faceted Clustering is a dynamic, user-controlled reorganization of a clustering, similar to a faceted navigation system. We report on evaluations using the AMBIENT corpus and demonstrate the potential of our approach by a comparison with two well-known clustering search engines.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Clustering* H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Search process* **Keywords:** search result clustering, cluster labeling

1. INTRODUCTION

Search result presentation strategies can be characterized as either being *relevance-based* or *diversity-based* [12]. While the objective of the relevance-based strategy is to serve the information need that is most likely associated with a submitted query, the objective of the diversity-based strategy is to serve different information needs in parallel. In this paper we propose to exploit the strengths of both result presentation strategies by combining the head of a ranked result list with a clustering of documents in the tail. This *competence partitioning strategy* entails fundamental differences compared to the current practice of Web search result clustering engines: Instead of adding an *alternative* means for search result inspection, we are aiming at a *complementing* clustering that takes the result list head into account. Our intuition is that users always have a look at the top results first, and turn to a result clustering only to avoid skimming through the result list tail. Taking

on this perspective allows us to derive two principles for the search result clustering task.

Principle 1: Avoid Shadowing. Users resort to a clustering of the tail only if their information need is still unsatisfied after having examined the result list head. I.e., it is unlikely that clusters containing documents similar to those in the head will improve user satisfaction. We term the undesired effect of topic repetition as *shadowing*. The shadowing effect becomes severe if the result list head is dominated by a single query aspect. If the head is rather diverse, users might be happy with a clustering that delivers additional resources for these aspects. With respect to the employed clustering algorithm, avoidance of shadowing means to not blindly cluster the results with the highest ranks, but to carefully select results from the tail.

Principle 2: Avoid Extreme Clusterings. Users who refuse to search the tail of a ranked result list won’t spent time on parsing an overlong list of cluster labels either, nor will they skim through the documents in a very large cluster. We hence constrain both the number of cluster labels and the maximum cluster size of a clustering to the size of the result list head. Here we take ten as value for both. I.e., our second principle limits the total number of documents that fit into a flat clustering to 100. This number may be increased by a hierarchical clustering, which, however, comes at the price that valuable search time is spent in unfolding and exploring the hierarchical structure.

With Faceted Clustering, we introduce a clustering algorithm tailored to the competence partitioning strategy and the two principles above. The algorithm generates a so-called monothetic clustering for a result list: A cluster C_l with label l is comprised of exactly those documents of a result list which contain l . Monothetic clusterings are preferred over polythetic clusterings in Web search result presentation because the cluster labels can be easily interpreted: they behave similar to adding an extra term to the original query. In order to comply with Principle 1, our clustering algorithm prefers to select cluster labels which do not appear in the result list head but which occur frequently in the result list tail. In order to comply with Principle 2, we introduce for a cluster C its *visible portion* as the ten most relevant documents in C .

To optimize the number of documents that fit into the clustering without violating Principle 2, we propose a cluster layout that can be considered in between the flat and the hierarchical layout paradigm—the *faceted layout*. The faceted layout allows to combine a set of cluster labels $L = \{l_1, \dots, l_k\}$ to form a new cluster C_L that contains the documents of $C_{l_1} \cap \dots \cap C_{l_k}$. Note that the visible portion of this intersection can reveal documents that are not visible in any of the clusters C_{l_1}, \dots, C_{l_k} . A user can explore visible documents by selecting and deselecting cluster labels in an ad-hoc fashion—similar to a faceted navigation system, which is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM’12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...\$10.00.

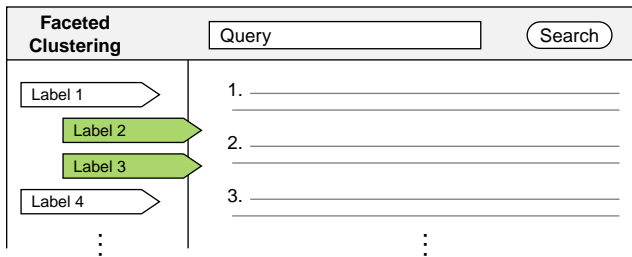


Figure 1: Abstract user interface of a Faceted Clustering search engine. Clicking on a cluster label will toggle the label and indicate its selection (green labels 2, 3) or deselection (labels 1, 4). Intersections between clusters are formed intuitively by selecting multiple cluster labels.

why we term our approach Faceted Clustering. The faceted nature introduces great flexibility and a strong means to individual searches: the number of clusters that can be built from ten cluster labels grows from 10 to 2^{10} . At the same time, the number of cluster labels to examine does not increase over the flat cluster layout; even the navigational overhead of unfolding and collapsing a hierarchical cluster structure is avoided. Figure 1 sketches the interaction principles of a user interface for Faceted Clustering.

2. RELATED WORK

In the following, selected entry points and recent results for Web search result clustering and related research fields are given.

Web Search Result Clustering. The existing approaches for Web search result clustering can be distinguished along three dimensions [2]: data-centric, description-aware, and description-centric. *Data-centric approaches* give top priority to clustering, while the formation of cluster labels has no effect on the partitioning of the snippets. The labels are usually derived from a cluster’s mathematical representation, e.g., a centroid or a medoid under a bag-of-words model. Hence the generated labels form a sequence of probably unrelated words, often lacking understandability. Examples for such approaches are WebCat [6] and AIssearch [13]. *Description-aware approaches*, by contrast, interweave the processes of clustering and labeling. In the existing approaches of this type, such as Grouper [17] or SnakeT [4], the labeling process controls a polythetic cluster analysis. *Description-centric approaches*, finally, employ cluster analysis solely for the purpose of discovering topics in a collection. Snippet partitioning is achieved by monothetic clustering, where each feature is used in an isolated manner to partition a collection into (overlapping) clusters. This approach leads to an improved understandability, which is bought by an possibly acceptable decrease in the clustering quality. Examples for such approaches are CAARD [7], Lingo [9], DisCover [8], and KeySRC [1]. Also the presented Faceted Clustering algorithm belongs to this category.

Cluster Labeling. The salient properties of good cluster labels are comprehensibility, descriptiveness, and discriminative power [13]. The basic building blocks of labels are phrases, whereas noun phrases [11], named entities [14], and title phrases [5] are recently discussed alternatives to improve comprehensibility. Descriptiveness means that a label should speak for each document in a cluster, while discriminative power means that the semantic overlap of two labels should be minimum.

Faceted Search. Faceted search can be considered as a “controlled, multi-view cluster analysis”. Without doubt, facets excel in satisfying the three label properties (comprehensibility, descriptiveness,

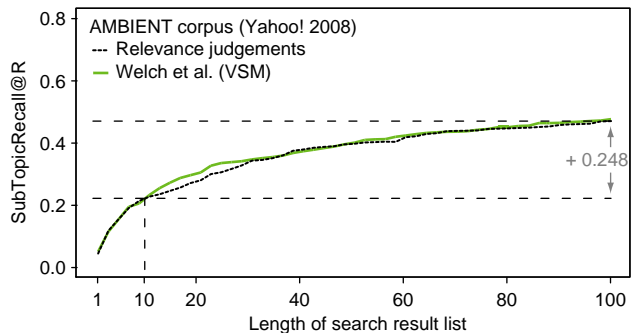


Figure 2: Average subtopic recall for the 44 topics of the AMBIENT corpus. The black curve shows the human assessment, the green curve shows the assessment by Welch’s unsupervised algorithm.

discriminative power) mentioned above. The automatic and ad-hoc generation of facets is an active research field, but currently this problem must be considered as unsolved [15]. With Faceted Clustering, we contribute an innovative approach to automatic but partly restricted facet generation: the semantic relationship between facet values is not considered—a criteria that is usually applied in faceted search applications.

3. THE WISDOM OF THE TAIL

The basic hypothesis of our research is that the result list tail is a valuable information source for a significant number of hidden query aspects. Faceted Clustering shall expose this wisdom to the user. This section gives evidence for this hypothesis by demonstrating that more than 20% of all query aspects can be found only in the tail of search result lists.

3.1 Query Aspect Detection in Search Results

Let $D_q = d_1, \dots, d_n$ be the ranked list of n search results retrieved for a query q . Each $d \in D_q$ denotes a result snippet, i.e., the concatenation of the search result’s title and its short description. In addition, for each query q a set A_q of query aspects is given. A query aspect is represented either by a short textual description, by a single Web page, or by multiple relevant Web pages obtained from individual searches. We denote the set of all query aspects of q that are covered by d as $coverage(d, A_q)$. To measure the query aspect recall for a result list D_q at rank R based on $coverage(d, A_q)$, we utilize the measure *subtopic recall* introduced by Zhai et al. [18]. Subtopic recall quantifies the fraction of query aspects covered by the first R search results:

$$SubTopicRecall@R = \frac{|\cup_{i=1}^R coverage(d_i, A_q)|}{|A_q|}, \quad d_i \in D_q.$$

The definition of *coverage* is crucial since it has to resemble human relevance judgments. A suitable corpus for subtopic analysis is the AMBIENT corpus [3], which has been used in numerous evaluations. The corpus contains 44 queries created on the basis of Wikipedia disambiguation pages that describe on average 18 diverse meanings (here: query aspects) of the query term. For each query the corpus provides 100 search results retrieved by Yahoo! in 2008. Each result is manually labeled with the query aspect that it covers. Figure 2 illustrates the average subtopic recall of the Yahoo! retrieval results; the dark curve shows the coverage assessments on the basis of human relevance judgments.

We would like to draw attention to two observations. (1) The search results retrieved from Yahoo! in 2008 cover 48.3% of the

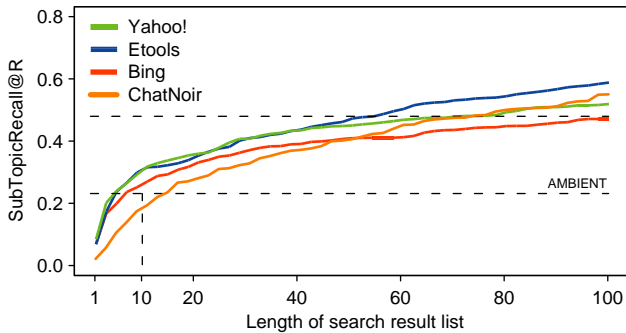


Figure 3: Average subtopic recall for the queries of the AMBIENT corpus. For each query 100 search results are retrieved from the four search engines Bing, Yahoo!, Etools, and ChatNoir.

query aspects listed on Wikipedia’s disambiguation pages. If we define the result list head $H \subset D_q$ to comprise the first ten search results, it covers 22.9% of all query aspects. Additional 24.8% of the query aspects are covered solely in the result list tail, which substantiates our hypothesis of “the wisdom of the tail” in search results. (2) Published in 2008, the retrieval results from Yahoo! may be suspected of being outdated. Recently Welch et al. introduced an unsupervised strategy to assess the $coverage(d, A_q)$ on the basis of the set of Wikipedia articles that are referenced on a disambiguation page [16]. The authors propose a normalized probability score $\Pr(a|d)$, which is computed for each query aspect a and document d such that $\sum_{i=1}^{|A_q|} \Pr(a_i|d) = 1$. If $\Pr(a|d) \geq 0.3$, a value which is not further motivated by Welch, then a is said to be covered by d . To actually compute $\Pr(a|d)$, the authors apply a vector space model and cosine similarity. In Figure 2 the green curve illustrates the average subtopic recall of Welch’s method computed for the queries from Yahoo!. A robust agreement with the manual relevance judgments can be clearly identified. Based on the impressive performance of Welch’s method on the AMBIENT corpus, one may well assess the subtopic coverage of the result lists from other search engines.

We apply Welch’s method to the search result lists obtained from the four search engines Bing, Yahoo!, Etools, and ChatNoir [10]. In contrast to the well-known search engines Bing and Yahoo!, Etools is a meta search engine and aggregates results of Bing and Yahoo! among others. Furthermore, our research group has developed an index for the ClueWeb09-T09B corpus, called ChatNoir, which utilizes the field-oriented retrieval model BM25F, PageRank, and SpamRank. For all queries of the AMBIENT corpus we retrieve the top 100 results of each search engine. The obtained average subtopic recall is illustrated in Figure 3. Like Yahoo! in 2008, also the modern search engines show the importance of the tail as a valuable information source. A tailored clustering approach that organized just the remaining query aspects in the long tail would perfectly complement the result list head and enable a more efficient and effective result analysis. The key to such a complementing behavior lies in the alleviation of undesired topic repetition, called shadowing.

3.2 The Shadowing Effect

To quantify the shadowing effect in search result lists for the AMBIENT dataset we compare the top-ten results H to the search results in the tail $D_q \setminus H$. Following Welch’s method for query aspect detection, we compute the cosine similarity $\varphi_{\cos}(d, d_i)$ with

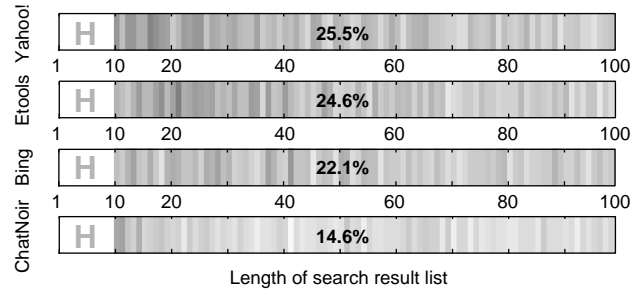


Figure 4: Average shadowing for the queries of the AMBIENT corpus, caused by the search engines Yahoo!, Etools, Bing, and ChatNoir. A bar indicates the shadowing at the result list positions 10...100. The darker a bar the more shadowing occurred at that position.

$d \in H$ and $d_i \in D_q \setminus H$. A search result d_i is said to be *shadowed*, iff $\max\{\varphi_{\cos}(d, d_i) \mid d \in H, d_i \in D_q \setminus H\} \geq 0.3$.

Figure 4 shows the results of our analysis. At each search result position the shadowing effect is encoded as a gray scale value between 0.0 (no shadowing) and 1.0 (always shadowed). I.e., the gray scale value reflects the relative frequency of shadowing events at a specific rank. Obviously, up to 25.5% of the search results retrieved by Yahoo! are shadowed. Compared to all other search engines, ChatNoir reveals the lowest shadowing of only 14.6%. Nonetheless, all studied search engines confirm the hypothesis that query aspects covered by the result list head reappear frequently throughout the tail. The Faceted Clustering algorithm, which is introduced next, penalizes documents that cause shadowing and hence is able to effectively alleviate shadowing in the generated clusterings.

4. FACETED CLUSTERING

Faceted Clustering aims at complementing a result list head with a tailored clustering of the result list tail. The algorithm belongs to the class of description-centric document clustering approaches, whose strategy may be summarized as “Description comes first”: Before the actual cluster formation process starts, a set of candidate labels \hat{L} is extracted from the document set D . Due to the monotonic nature of this formation process, a label $l \in \hat{L}$ induces—in the sense of: determines—for a set D a cluster $C_l \subseteq D$. Similarly, a label *set* L induces a *set* of clusters \mathcal{C} , also termed a clustering. Note that the extraction of appropriate cluster labels is a non-trivial task that involves the application of sophisticated natural language processing techniques like noun phrase extraction and morphological generalization. The paper in hand won’t focus nor contribute to the label extraction problem; in our experiments we resort to state-of-the-art technology.

4.1 Cluster Formation

Given a set of candidate labels \hat{L} , cluster formation means to construct a label subset $L \subset \hat{L}$ whose induced clustering \mathcal{C} adheres to the two principles of the competence partitioning strategy. Such a clustering minimizes shadowing and avoids extreme clusterings (i.e., a high cluster number as well as large clusters), and, at the same time, maximizes the number of visible documents. Similar to other description-centric document clustering algorithms, Faceted Clustering pursues a greedy strategy for cluster formation. The algorithm is outlined in Algorithm 1.

In each of the k iterations a cluster label $l^* \in \hat{L}$ is chosen that maximizes *candidateScore*. The label l^* is added to the set of already selected labels L , and the intermediate clustering \mathcal{C} as well as the document sets N (not yet visible) and U (untouched) are

Algorithm 1 Faceted Clustering

Input: Candidate labels \hat{L} , Documents D
Parameters: Number of iterations k
Output: Labels L , Clustering \mathcal{C}

$L = \emptyset, \mathcal{C} = \emptyset, N = D, U = D$
for $r = k$ **downto** 1 **do**
 $l^* = \operatorname{argmax}_{l \in \hat{L}} (\text{candidateScore}(\mathcal{C}, l, r, N, U))$
 $L \leftarrow L \cup \{l^*\}$
 $\mathcal{C}, N, U \leftarrow \text{updateClustering}(\mathcal{C}, l^*, N, U)$
end for
return (L, \mathcal{C})

updated with regard to l^* . Note that by means of set intersections up to 2^{k-r} new clusters could be added to \mathcal{C} in each round, but we restrict the number of operands (clusters) within an intersection to three. Alternatively, the update of \mathcal{C} can be bound to a constant size. Documents that are assigned to a cluster are marked as *touched* and removed from the set of untouched documents U . Recall that only the k most relevant documents in a cluster will be visible to a user. In this regard the set N keeps track of the assigned but not yet visible documents. The overall runtime is in $O(|\hat{L}| \cdot k)$, which puts Faceted Clustering on the list of the fastest description-centric algorithms such as DisCover [8] and CAARD [7].

The heart of Faceted Clustering is the heuristic *candidateScore*, outlined in Algorithm 2. Given an intermediate clustering \mathcal{C} the heuristic estimates for a candidate label l (via the induced cluster C_l) the potential to boost the number of visible documents in the final clustering. This estimate is computed as a score of two summands whose rationale is as follows.

The first term, $|A|$, is the number of documents that will become additionally visible if the cluster C_l is added to the intermediate clustering \mathcal{C} . A is the intersection of N , the set of not yet visible documents, and V , the set of visible documents if \mathcal{C} is updated with C_l . I.e., the set V is union set of the intersections of C_l 's head and the clusters in \mathcal{C} , completed by C_l 's head. In the first iteration, the maximum value of $|A|$ equals k , whereas in the following iterations the number of documents in N and their distribution in the intermediate clustering is considered.

The second term quantifies the ‘‘potential’’ of a candidate label l . It considers the documents which have not yet been touched, but which become a non-visible cluster document after \mathcal{C} is updated with C_l . In this regard the set R contains all documents that the function *updateClustering* (see Algorithm 1) would remove from U but retain in N . Having many of these documents in different clusters is important since they entail a high score when forming the intersection sets. Loosely speaking, we are looking for labels that bring many new documents into play. The higher the potential of a label is, the more likely is the event that a future iteration will return a label that entails a high $|A|$ -value. Since the probability for this event decreases with each iteration, the weight of the second term is reduced in each round. For smoothing purposes, R is normalized by $|U \setminus A|$, which bounds the maximum value of the second term to r .

5. EVALUATION

This section reports on two experiments that analyze whether the Faceted Clustering algorithm optimizes Web search result clustering according to the competence partitioning strategy. In the first experiment we show that the faceted cluster layout leads to a significant increase in result coverage compared to a flat cluster layout that tries to avoid extreme clusterings. The second experiment re-

Algorithm 2 Faceted Clustering: *candidateScore*

Input: Clustering \mathcal{C} , Label l ,
Remaining iterations r , Not visible documents N ,
Untouched documents U
Output: Candidate score *score*

$V = \{d \mid d \in \text{head}(C_l \cap C), C \in \mathcal{C}\} \cup \text{head}(C_l)$
 $A = V \cap N$ /* Additionally visible under l */
 $R = (C_l \cap U) \setminus A$ /* Retained in N under l */
 $\text{score} = |A| + r \cdot |R| / |U \setminus A|$
return (score)

veals that, when being compared to well-known competitors, only Faceted Clustering is able to reduce shadowing within its clusters. Both experiments use search result lists that we received for the 44 queries of the AMBIENT from the search engine Yahoo!. The result lists contain 500 result snippets, from which we consider the first ten as the result list head.

5.1 Result Coverage

A key characteristic of the Faceted Clustering approach is its flexibility to assign documents to clusters, while avoiding extreme clusters and navigational overhead at the same time. However, the question is whether the term distribution of the query results in fact allow an exploitation of the gained space. To evaluate the benefit of Faceted Clustering in this regard, we cluster the Yahoo! result lists with both the Faceted Clustering algorithm and an algorithm variation that generates a flat clustering. If the number of results that are covered by faceted clusterings is significantly higher than for flat clusterings, we have a strong indication for the superiority of Faceted Clustering.

For the Flat Clustering algorithm, the score of a label l is based only on the number of new results the cluster C_l would add to an intermediate clustering \mathcal{C} . Intersections with existing clusters are not considered. The quantity k , which controls the maximum number of documents for a cluster, as well as the amount of cluster labels $|L|$ is set to ten for both algorithms. I.e., for Flat Clustering the result coverage is bound to 100.

Before applying the clustering algorithms, the set of candidate cluster labels \hat{L} has to be determined. This step is independent of the cluster formation step, and many different strategies have been proposed for this purpose in the past (cf. Section 2). Here, \hat{L} is the set of all unigrams and bigrams which occur in the result snippets and which contain no common stopwords. Compared to labels of more sophisticated approaches, unigrams and bigrams are shorter on average and hence are better suited for the purpose of forming cluster intersection.

The result of this experiment is shown in Figure 5 in the form of two histograms. Note that the histograms do not overlap at all, which is because the Faceted Clustering approach consistently surpasses the maximum result coverage of plain clusterings. With 130.7 results on average, the Faceted Clustering approach is clearly superior to the baseline approach, which yields an average result coverage of 89.8 only. According to the Wilcoxon signed-rank test, the performance difference is significant at confidence level 0.01.

The achieved result coverage gain results from the possibility to form cluster intersections: From the 5755 unique results covered in total by all 44 clusterings, 40% are covered only by intersecting two clusters, and 12% are covered only by intersecting three clusters. In 0.2% of all cases, even four intersections are needed. We infer from our result coverage experiment that the faceted clustering layout is highly effective when it comes to generate well-balanced, non-extreme clusterings.

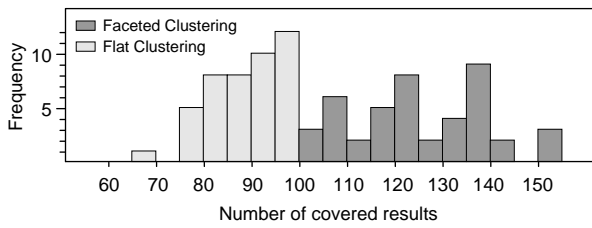


Figure 5: The two histograms show the frequency at which the Flat Clustering algorithm and the Faceted Clustering algorithm achieve a certain result coverage. Basis are the search result clusterings for the 44 queries of the AMBIENT corpus.

5.2 Shadowing Reduction

A second key characteristic of Faceted Clustering is the alleviation of the shadowing effect in its generated clusterings. As a baseline for the evaluation we take the average shadowing in the original result lists from Yahoo!, which is 25.5% (cf. Section 3 and Figure 4). We expect the average shadowing under Faceted Clustering to be moderately lower than for the baseline. A dramatic decrease cannot be expected since the Faceted Clustering algorithm enfolds its full power in cases where the result list head is dominated by a single query aspect. To further substantiate our evaluation, we also measure shadowing for the clusterings generated by two other well-known description-centric clustering algorithms. The first algorithm, Lingo, uses singular value decomposition to find topically related terms and groups them under the same cluster label. The second algorithm, STC, uses a suffix tree representation of the documents to find reoccurring phrases that can be used as cluster labels. Both algorithms are contained in the Carrot2 software library.¹

To apply the shadowing quantification method from Section 3 to a clustering, we take the documents of each cluster and add them to the head of the relevant result list one at a time. To determine the order in which the clusters' documents are added, we use the relevance scores that are provided by STC and Lingo. For Faceted Clustering, we added the documents in the order of their removal from the set N of not yet visible documents. The results of our evaluation are shown in Figure 6: The Faceted Clustering algorithm successfully reduces shadowing from 25.5% in the original list to 18.9% in the clustering. This decrease is significant according to a proportions test at confidence level 0.01.² By contrast, Lingo and STC, which both do not take the shadowing effect into account, cause a slight increase in shadowing.

6. CONCLUSIONS

With Faceted Clustering, we present innovative technology to make the most out of the critical time and size constraints users impose on the analysis on Web search results. Besides its possibility to operationalize a competence partitioning strategy for Web Search, we see Faceted Clustering as a general means to effectively explore large text collections.

7. REFERENCES

- [1] A. Bernardini, C. Carpineto, M. D'Amico. Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering. *Proc. of WI-IAT 2009*, pp. 206-213.
- [2] C. Carpineto, S. Osiński, G. Romano, D. Weiss. A Survey of Web Clustering Engines. *ACM Comp. Surveys 2009*, Art. 17.

¹<http://project.carrot2.org/>

²`prop.test` from the R stats package.

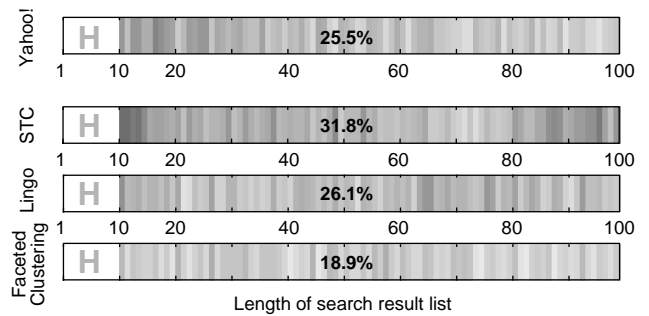


Figure 6: Average shadowing for the queries of the AMBIENT corpus, caused by the search engine Yahoo! (copied from Figure 4) and the clustering algorithms STC, Lingo, and Faceted Clustering.

- [3] C. Carpineto, G. Romano. AMBIENT corpus. <http://credo.fub.it/ambient>.
- [4] P. Ferragina, A. Gullí. A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. *Proc. of WWW 2005*, pp. 801-810.
- [5] F. Geraci, M. Pellegrini, M. Maggini, F. Sebastiani. Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution. *Proc. of SPIRE 2006*, pp. 25-36.
- [6] F. Giannotti, M. Nanni, D. Pedreschi, F. Samaritani. WebCat: Automatic Categorization of Web Search Results. *Proc. of SEBD 2003*, pp. 507-518.
- [7] K. Krishna, R. Krishnapuram. A Clustering Algorithm for Asymmetrically Related Data with Applications to Text Mining. *Proc. of CIKM 2001*, pp. 571-573.
- [8] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, R. Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. *Proc. of the WWW 2004*, pp. 658-665.
- [9] S. Osiński, J. Stefanowski, D. Weiss. Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. *Proc. of IIPWM 2004*, pp. 359-368.
- [10] M. Potthast, M. Hagen, B. Stein, J. Graßegger, M. Michel, M. Tippmann, C. Welsch. ChatNoir: A Search Engine for the ClueWeb09 Corpus. *Proc. of SIGIR 2012*.
- [11] J. Stefanowski, D. Weiss. Comprehensible and Accurate Cluster Labels in Text Clustering. *Proc. of RIAO 2007*, pp. 198-209.
- [12] B. Stein, T. Gollub, D. Hoppe. Beyond Precision@10: Clustering the Long Tail of Web Search Results. *Proc. of CIKM 2011*, pp. 2141-2144.
- [13] B. Stein, S. Meyer zu Eißén. Topic Identification: Framework and Application. *Proc. of I-KNOW 2004*, pp. 353-360.
- [14] H. Toda, R. Kataoka. A Clustering Method for News Articles Retrieval System. *Proc. of WWW 2005*, pp. 988-989.
- [15] D. Tunkelang. *Faceted Search*. Morgan & Claypool Publishers, 2009.
- [16] M. Welch, J. Cho, C. Olston. Search Result Diversity for Informational Queries. *Proc. of WWW 2011*, pp. 237-246.
- [17] O. Zamir, O. Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. *Proc. of WWW 1999*, pp. 1361-1374.
- [18] C. Zhai, W. Cohen, J. Lafferty. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. *Proc. of SIGIR 2003*, pp. 10-17.