Technical Report

Evaluation Methods for Rankings of Facetvalues for Faceted Search

Anne Schuth and Maarten Marx

ISLA, University of Amsterdam, The Netherlands
{anneschuth,maartenmarx}@uva.nl

Abstract. We introduce two metrics aimed at evaluating systems that select facetvalues for a faceted search interface. Facetvalues are the values of meta-data fields in semi-structured data and are commonly used to refine queries. It is often the case that there are more facetvalues than can be displayed to a user and thus a selection has to be made. Our metrics evaluate these selections based on binary relevant assessments for the documents in a collection. Both our metrics are based on Normalized Discounted Cumulated Gain, an often used Information Retrieval metric.

To appear in the *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation 2011.*

1 Introduction

Search interfaces to semi-structured data often provide ways of refining a full-text search query by selecting values of meta-data fields. These fields —called *facets*— and their values —called *facetvalues*— are then used to filter the results.

We can usually only present a limited number of such facetvalues to a user; both because we have limited amount of space (on a screen) but also because we do not want to put a burden of sifting through a large amount of facetvalues on a user. So, out of all facetvalues a selection has to be made; this paper investigates ways of evaluating such a selection.

In broad terms, we aim at finding a metric that prefers facetvalues that would minimize navigation for a user; a metric that prefers the shortest navigational path through the collection of documents. We want to guide a user in as little as possible steps to all documents that are relevant to his query.

We view the setting in which the selection of facetvalues occurs as follows. We have a collection of documents, some queries and binary relevance judgments (by human assessors) for some documents in the collection for each query, we assume all other documents irrelevant. Besides, we assume that a query defines

a strict linear order over the documents.[1] This ordering we assume given. So, for each query we know which documents are relevant and how all documents should be ordered. Also, all our documents are semi-structured, meaning that they contain textual data (on which the ordering is based) as well as meta-data. This meta-data determines to which facetvalues a document belongs.

2 Motivation

While there has been work on evaluation faceted search systems from a user interface perspective (Kules et al., 2009; Burke et al., 1996; English et al., 2002; Hearst, 2006, 2008, 2009), no work has focused on ranking facetvalues from an Information Retrieval perspective. We view our work as enabling research in that direction and would propose doing so in an evaluation campaign with a setup as described below.

Task Each participant receives the following: *a*) a collection of queries Q; *b*) a collection of semi-structured documents D; *c*) a strict linear order defined on these documents for each query $q \in Q$; and *d*) a set of facets that may be used, the corresponding facetvalues are dictated by the structured part of the document collection. The task is then to return an ordered list or —depending on the measure— tree of facetvalues that maximizes one of the two metrics defined in this report.

Evaluation Both our metrics, as described in Section 3, can use simple binary relevance judgments on document per query level. And both return a single number that measures how good a list or tree of facetvalues is. This can be averaged over all queries. To evaluate a participant, the following is needed: *a*) a collection of semi-structured documents D; *b*) a strict linear order defined on these documents for each query $q \in Q$; *c*) binary relevance judgments for each document $d \in D$, for each query $q \in Q$; and *d*) the submission of the participant: a list or tree of facetvalues for each query $q \in Q$. [2]

3 Two Evaluation Metrics

We introduce two new evaluation metrics. First the Normalized Discounted Cumulated Gain which is an adaption of an existing metric NDCG as described by Järvelin and Kekäläinen (2002). Second, we introduce a new metric called NRDCG which is recursive version of NDCG. Each of our metrics is meant to measure the quality of an *ordered* list or tree of facetvalues. In Table 1 we first introduce some notation, partly inspired by Dash et al. (2008).

[1] Such an ordering can be based on some similarity score between textual data of the document and the query

[2] Note that these requirements imply that at least the INEX 2010 Data Centric Track (Trotman and Wang, 2010) data and relevance judgments can be used with almost no adjustments.

Table 1. Notation used for the definition of our metrics, inspired by Dash et al. (2008).

d	a document, consisting of triple $< t, FV, R >$. With free-text t, set of facetvalues FV and a set of relevance judgments R consisting of $r_q \in \{0, 1\}$ for each query $q \in Q$, where $r_q = 1$ if the document is judged relevant to query q by human judges.
D	list of documents in arbitrary order
D_q	list of documents D ordered by query q
D_f	list of documents D filtered by facetvalue f (in arbitrary order). Or $D_f = \{d : d \in D \wedge f \in FV(d)\}$
f	a facet value pair *facet:value*. A facetvalue is a property of a document, in filtering operations it preserves only those documents that have this property.
F	list of facetvalues.
FT	tree of facetvalues.
q	a full-text query that can define an ordering on D
Q	a set of full-text queries
$D_q[i \ldots j]$	list of documents i up to j in the ordered list of documents D_q. *Note that the result of $D[\cdot]$ and $D_f[\cdot]$ is arbitrary since the order of those lists is arbitrary.*
$t(d)$	the free-text t of document d
$FV(d)$	the set of facetvalues FV of document d
$r(d, q)$	the binary relevance judgment r of document d with respect to query q
$R(D, q)$	list of the relevant documents given a query q that occur in list of documents D. Or $R(D, q) = \{d : d \in D \wedge r(d, q) = 1\}$
n	maximum number of facetvalues per navigation step
p	maximum number of resulting documents in which to look for relevant documents
λ	used in NRDCG to balance direct Gain with Gain in drill-down, $\lambda = 0$ causes NDRCG to reduce to NDCG. $0 \leq \lambda < 1$.

3.1 Normalized Discounted Cumulated Gain

This metric focuses on the following two rather loosely formulated aspects: *a*) prefer facetvalues that would return a lot of relevant documents high in the result list; and *b*) prefer facetvalues that would return relevant documents we have not seen yet by earlier facetvalues. The first aspect can be measured by counting the number of relevant documents end up in the top p of results if the document-list D_q were filtered by a facetvalue. The second aspect can then be satisfied by discarding all relevant documents that were already covered by earlier facetvalues, in a ranked list (or tree) of facetvalues. Our notion of Gain, as explained below, is designed to capture both aspects. Because we use a discounted measure we value the Gain of a facetvalue more if it is returned earlier in a ranked list of facetvalues. If we allow the (binary) relevance judgments for documents per query to transfer to facetvalues, we get *graded* relevance for facetvalues. We want these relevance judgments —these gains— to reflect the number of relevant documents in the top p result if a facetvalue were selected. Then, to judge the quality of a ranked list of facetvalues, we could use the Normalized Discounted Cumulated Gain (NDCG) measure (Järvelin and Kekäläinen, 2002) that is designed to evaluate a ranking using graded relevance.

In order to be able to compare DCG for several queries, a normalization step is needed. We can use the regular version of Normalized Discounted Cumulated Gain:

$$NDCG(D, F, q) = \frac{DCG(D, F, q)}{IDCG(D, q)} \tag{1}$$

With a regular definition of Discounted Cumulated Gain:

$$DCG(D, F, q) = \sum_{i=1}^{\min(n,|F|)} \frac{G(D, F, i, q)}{\log_2(i + 1)} \tag{2}$$

So far, nothing was new. Only our definition of Gain is adjusted to reflect the transfer of relevant judgments of documents to facetvalues. We define Gain, $G(D, F, i, q)$ as follows:

$$G(D, F, i, q) = \left| R(D_{q,f_i}[1 \dots p], q) \setminus \bigcup_{j=1}^{i-1} R(D_{q,f_j}[1 \dots p], q) \right| \tag{3}$$

Note how this version of Gain does not take relevant documents covered by earlier facetvalues into account; nothing is gained by returning the same relevant result more than once. It forces the overall measure to prefer facetvalues that cover *new* relevant documents.

Evidently, for the normalization step, we need to calculate how well an ideal ranked list of facetvalues for this query would do; we calculate the Ideal Discounted Cumulated Gain as follows:

$$IDCG(D, q) = \sum_{i=1}^{n} \frac{IG(D, i, q)}{\log_2(i+1)} \tag{4}$$

Where our version of the Ideal Gain states that each ith facetvalue could cover at most p new relevant document:

$$IG(D, i, q) = \max(0, \min(p, |R(D, q)| - (i-1) \cdot p)) \tag{5}$$

Since we normalized the measure for each query, averaging is simple:

$$\overline{NDCG}(D, FT, Q) = \frac{1}{|Q|} \cdot \sum_{q \in Q} NDCG(D, FT, q) \tag{6}$$

3.2 Normalized Recursive Discounted Cumulated Gain

We could also look at the problem of finding the right facetvalues differently. In a real (and possibly even optimal) system a user might have to navigate through multiple facetvalues before he arrives at the desired (relevant) document. In other words, that means that it might take a couple steps before all *relevant* documents end up in the top p results. If we are trying to optimize this navigation we might want to take into account the consecutive facetvalues —and their quality— that a user encounters in a navigation session.

So, instead of looking for the optimal ranked *sequence* of facetvalues, we are looking for an optimal ranked *tree* of facetvalues. We change the setting described in the introduction; we now look for a facetvaluetree FT that optimizes our recursive metric. Such a facetvaluetree FT essentially consists of nested lists of facetvalues and looks like this:

$$FT = (f_1(FT_1), \dots, f_n(FT_n)) \tag{7}$$

For a tree FT we denote the children of the root with f_i, and we use the notation FT_i to denote the subtree rooted at f_i. The only (natural) restriction on this tree is that paths may not contain a facetvalue more than once.[3]

We define a metric to evaluate this tree in a fashion similar to NDCG, but defined recursively and thus called: Normalized Recursive Discounted Cumulated Gain.

$$NRDCG(D, FT, q) = \frac{RDCG(D, FT, q)}{IRDCG(D, q)} \tag{8}$$

We use a recursive version of DCG called Recursive Discounted Cumulated Gain, that is not different except for that it sums up a Recursive Gain.

$$RDCG(D, FT, q) = \sum_{i=1}^{\min(n,|FT|)} \frac{RG(D, FT, i, q)}{\log_2(i+1)} \tag{9}$$

The Recursive Gain, RG, is a mixture model that is composed of a direct gain borrowed from the normal $NDCG$ and a recursive step. Note that for the recursive call we shrink the document set with those that were displayed already; this causes the measure to focus on unseen (relevant) documents in the remaining steps of a drill-down session.

$$RG(D, FT, i, q) = (1-\lambda) \cdot G(D, FT, i, q) + \lambda \cdot RDCG(D \setminus D_{q,f_i}[1 \dots p], FT_i, q) \tag{10}$$

Setting $\lambda = 0$ reduces the measure to NDCG. Setting $\lambda = 1$ would lead to a zero score (and even division by 0) thus this is not allowed. If $\lambda > 0.5$, the recursive part would get more weight, thereby preferring relevant documents to appear later in a drill-down session. Given that we are after a quick navigation session, we suggest setting $0 < \lambda < 0.5$.

Note that no explicit stopping criteria is needed as $G(\cdot)$ returns 0 for empty document lists and $RDCG(\cdot)$ returns 0 for empty facetvalue lists.

Also, verify that we can indeed simply use $G(D, FT, i, q)$ —even though that function is defined on a list—, as the Gain function is only looking at the children f_i and f_j of the root of FT. Documents (relevant or not) covered by ancestor facetvalues are filtered out by the recursive call to $RDCG(\cdot)$, that is done using $D \setminus D_{q,f_i}[1 \dots p]$ instead of simply D.

To normalize the $RDCG$, we will need an ideal version, Ideal Recursive Discounted Cumulated Gain ($IRDCG$), which naturally is defined recursively:

$$IRDCG(D, q) = \sum_{i=1}^{\min(n,|R(D,q)|)} \frac{IRG(D, i, q)}{\log_2(i+1)} \tag{11}$$

$$IRG(D, i, q) = (1-\lambda) \cdot IG(D, i, q) + \lambda \cdot IRDCG(D \setminus R(D, q)[1 \dots p], q) \tag{12}$$

As with $NDCG$, we average over all queries to arrive at $\overline{NRDCG}(D, FT, Q)$.

[3] This restriction is needed because we only filter out the top p results in the recursive call to $RDCG$, and not *all* documents that are covered by a facetvalue

4 Conclusions

We have have introduced two related measures for evaluating rankings of facetvalues. One might prefer NDCG over NRDCG for its simplicity while the recursive variant might be preferred because it is much more fine grained. Meaning that NRDCG is better at judging a selection of facetvalues were the number of relevant documents is small and harder to retrieve.

Even though we did not include experimental results, our experiments have shown that both measures rank systems that are expected to perform better higher. That is a necessary —not sufficient— indication that our measures perform as intended. Future work should look into proper evaluation of the introduced metrics. The next opportunity to do so will be the INEX 2011 Data Centric Track on Faceted Search where our metrics will be used for evaluation. We expect that either or both of our evaluation metrics will foster the development of systems that focus on strategies of selecting the right facetvalues.

Acknowledgments Maarten Marx acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599. This research was also supported by the Netherlands organization for Scientific Research (NWO) under project number 380-52-005 (PoliticalMashup).

Bibliography

Burke, R. D., Hammond, K. J., and Young, B. C. (1996). Knowledge-based navigation of complex information spaces. In *Proceedings of The National Conference On Artificial Intelligence*, volume 462, page 468.

Dash, D., Rao, J., 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 mining - CIKM '08*, page 3, Napa Valley, California, USA.

English, J., Hearst, M., Sinha, R., Swearingen, K., and Yee, K. P. (2002). Hierarchical faceted metadata in site search interfaces. In *CHI'02 extended abstracts on Human factors in computing systems*, page 628639.

Hearst, M. (2006). Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR Workshop on Faceted Search*, page 15.

Hearst, M. (2008). Uis for faceted navigation: Recent advances and remaining open problems. In *Proc. 2008 Workshop on Human-Computer Interaction and Information Retrieval*.

Hearst, M. (2009). *Search user interfaces*. Cambridge Univ Pr.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20:422446. ACM ID: 582418.

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 '09, page 313–322, New York, NY, USA. ACM. ACM ID: 1555452.

Trotman, A. and Wang, Q. (2010). Overview of the inex 2010 data centric track.