High Accuracy Recall Task

Andrew Trotman
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

Surya Kallumadi
Kansas State University
Kansas,USA
surya@ksu.edu

Jon Degenhardt
eBay inc.
California, USA
jondegenhardt@gmail.com

ABSTRACT

We identify a new information retrieval task for eCommerce that we call the high accuracy recall task. That task is to identify as many relevant documents, and as few non-relevant documents as possible, such that regardless of the rank ordering, the precision remains high.

We demonstrate a need to investigate this problem, we propose metrics to measure the quality of the results, and we suggest how a document collection might be built and queries might be generated.

CCS CONCEPTS

• **Information systems → Retrieval effectiveness**;

KEYWORDS

eCommerce, Performance Metrics, Quantitative Analysis

ACM Reference Format:
Andrew Trotman, Surya Kallumadi, and Jon Degenhardt. 2018. High Accuracy Recall Task. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2018 eCom).* ACM, New York, NY, USA, 5 pages. https://doi.org/

1 INTRODUCTION

eCommerce search engines often provide multiple rank orders of the results. Amazon, for example, offers the user 6 orders ranging from "Relevance" to "Avg. Customer Review" and "Price: Low to High", Trademe offers the user a choice of 10 rank orders.

Search engine evaluation has traditionally been based on measuring the ability of the search engine to place relevant documents at the top of a results list. The working hypothesis is the probabilistic ranking principal – documents in a results list should be ranked in order of most probably relevant to the user, to least probably relevant to the user. For an eCommerce search engine its necessary to diverge from this principal because of the multiple rank orders.

In this short opinion piece we explore how we might evaluate the quality of an eCommerce search engine offering multiple rank orderings using Amazon and Trademe as running examples.

First we explore the search interface of these two sites and show that they, indeed, provide the user with the ability to re-sort the results of their query. We then examine the quality of the first page of results for a single query and show that the quality varies for different rank orderings. Indeed, when we examine the multiple orders for a single query we show that these search engines find it difficult to identify documents that are relevant to that one query.

We believe that the problem is a consequence of the quality of the set of documents[1] retrieved by the search engine (and then ranked). If this *recall base* contains many false positives then it is inevitable that some rank order (either known now, or future rank order) will place a non-relevant document high in the results list.

There are two ways we might measure the quality of the results. The first is to make no assumptions on the rank order and to measure the quality of the retrieved documents as a set – which we show is infeasible in a large collection. The second is to evaluate using the rank ordering the sites provide and we propose a metric to accomplish this.

The probabilistic ranking principal also fails for eCommerce because it assumes the user is trying to find a relevant document. In the case of a user browsing an eCommerce site to, for example, get a "feel" for the going price and quality of a used book, they are trying to compare the top few (k) results. We examine this search modality as a case of invested effort – something that has previously been examined as the expected search length (ESL) and tolerance to irrelevance (T2I). We introduce a metric that measures the proportion of non relevant documents the user will see when they reach the kth relevant document.

2 PROBLEM STATEMENT

Modern Internet search engines consist of a document collection and a sophisticated search engine that, given a user query, resolves the query against the collection to produce a list of results. The probabilistic ranking principal [11] states that the results should be presented in order of most likely to be relevant to least likely to be relevant.

The probabilistic ranking principal has been examined and questioned many times. Fuhr [6], for example, suggests that, in practice, it is not suitable for use in an interactive setting. Work at TREC [3] suggests that in a web setting with millions of document and ambiguous queries it is important to diversify results in a results list. For example, when searching for "Apple", the best result appears to contain results about Apple Inc., as well as Apple Corps., and the fruit. This ambiguity resolution is a natural part of the Wikipedia which has 61 links on the"Apple (disambiguation)" page, broken into 8 categories.[2]

The probabilistic ranking principal is directly questioned by the user interfaces to many eCommerce sites. Figure 1 (left) shows the 6 different sort orders on Amazon, ranging from "Relevance" to "Price: Low to High" to "Newest Arrivals". Of these 6, only 1 (Relevance) could be considered to be applying the probabilistic

[1]In eCommerce it is usual to use the term *document* to refer to a product listing – which may or may not contain reviews, ratings, and so on.
[2]https://en.wikipedia.org/wiki/Apple_(disambiguation), visited: 23 April 2018

SIGIR 2018 eCom, July 2018, Ann Arbor, Michigan, USA

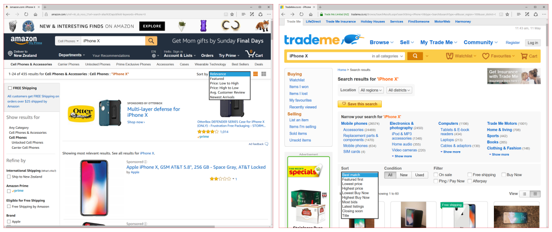

Figure 1: Amazon (left) and Trademe (right) result orderings for query "iPhone X"

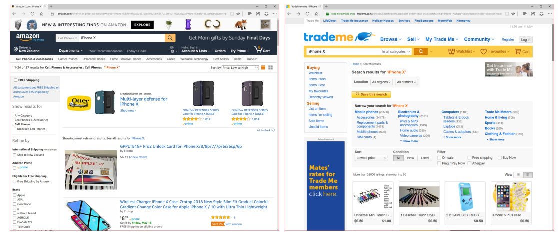

Figure 2: Amazon (left) and Trademe (right) price low to high results for query "iPhone X"

ranking principal. Figure 1 (right) shows the sort orders for Trademe, an Australasian eCommerce site and its 10 sort orders which, while not dissimilar to those of Amazon, also include "Most Bids", and "Title", neither of which are ordered by the probabilistic ranking principal. We note that title ordering has been examined by Sherlock & Trotman [13].

If most of the available rank orderings of eCommerce sites are not "Relevance", then evaluation of the search engine cannot be done on the assumption that it is. That is, the ability to put the most relevant document at the top of the results list is only one facet of rank orderings to be evaluated when measuring the quality of a site.

3 ALTERNATIVE RANK ORDERS

It has been posited that if the ranking function is effective enough then a few false positive documents in the results lists is acceptable because the ranking function will place those documents at the bottom of the list and no-one will see them [8]. This approach is, unfortunately, ineffective with sort orders based on constant document features (such as price).

To illustrate this point we searched for "iPhone X" on both Amazon and Trademe, and ranked using price low to high – something we consider entirely reasonable for a user to do and quite likely a high frequency (or head) query. While using a single query is far from evidence of a systematic problem, it can be considered to be a proof, by example, of the existence of a problem.

Figure 2 left shows the results for Amazon while Figure 2 right shows the results for Trademe. On Amazon, neither of the first two listings are for phones (and neither is the advertising). On Trademe, two are for a stylus, and two are for cases (but not for the iPhone X). On both Amazon and Trademe none of the results on the first page are for an iPhone X. When ordered by relevance, the top 4 results on both sites (the first page) are all iPhone X.

To demonstrate that this problem is not unique to "price low to high", we issued the same query on Amazon and looked at the top document of each of the sort orders and examined the top result. Of the 6 sort orders on Amazon, 3 failed to place an iPhone X at position 1. On Trademe only 2 of the 10 sort orders placed an iPhone X at position 1. A single query is insufficient to draw robust conclusions, but demonstrates the existence of a problem.

It is reasonable to conclude that the found document set (the recall base) contains false positives which in "Relevance" order are pushed low down in the results list, but in other sort orders these false positives can be presented to the user.

4 EVALUATION

The comparison between Amazon and Trademe shows that not only are there several possible sort orders, but that those orders are different between different sites. This suggests that it might not be possible to close the list of sort orders – in other words, Amazon might adopt some new sort orders in the future.

This raises the question how to evaluate the quality of a search engine in light of sort orders that have not yet been proposed, as well as those that have. We believe that this can be achieved by measuring the quality of the recall base rather than the ranking. The obvious measure is the F_1 of precision and recall, at least as far a buying is concerned. We explore this in section 4.1.

Information retrieval metrics are, in essence, models of users. We are aware of very little work examining user interaction on eCommerce sites (but see Sharma et al. [12]). We assume two models, buying and browsing.

When browsing the user wants to see k relevant documents to compare (for example) their colour, quality, age, and price. We explore metrics for browsing in section 4.3.

4.1 Buying: All Possible Orderings

The accuracy of a search engine irrespective of the rank order of the documents in the results list is given by the set-wise precision. Precision is defined as the proportion of documents that the search engine returns that are relevant.

$$p = \frac{f_r}{f}, \tag{1}$$

where f_r is the number of known-relevant documents retrieved by the search engine, and f is the number of documents in the results list. Problematically, a strategy for scoring high in set-wise precision is to return only one relevant document – which is clearly not in the interests of the user (unless there is only 1 relevant document in the collection).

A solution is to measure the recall, the proportion of the known relevant documents in the collection that the search engine returns to the user,

$$c = \frac{f_r}{r} \tag{2}$$

where c is the recall, f_r is the number of known-relevant documents retrieved by the search engine, and r is the number of known-relevant documents in the collection. Problematically, a strategy for scoring high in recall is to return all documents – which is not in the interests of the user because the precision can be expected to be low.

If both set-wise precision and recall are very high then the search engine has returned a large proportion of the relevant documents and putting them in any order should nearly satisfy the probability ranking principle. This is usually measured using the F_1 score, the harmonic mean of precision and recall. The F_1 score is rank-order invariant. That is, it is a good indicator of quality before the rank order is known. To compute F_1, its necessary to know r.

In a large document collection such as those at Amazon (about 550 million listings)[3] and Trademe (about 6 million listings)[4], for a given query, it isn't possible to know the number of relevant documents in the collection (items for sale that the user might want to purchase or browse). So computing set-wise recall is infeasible. We propose three solutions to this: random sampling, reordering, and pooling.

A random sample taken from the document collection could be used. We observe that there are two possible outcomes of a randomly selected document – either it is relevant or it is not – so the distribution is binomial and each randomly selected document is a Bernoulli trial.

Assuming the search engine is perfect ($precision = recall = 1$), we have an estimate of the number of relevant documents in the collection is given by:

$$\hat{p} = \frac{f_r}{N}, \tag{3}$$

where $\hat{p}$ is the estimated proportion of the collection that is relevant, f_r is the number of found documents, and N is the collection size.

The confidence we have in that estimate is

$$\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \tag{4}$$

Allowing for a confidence interval of 10% of $\hat{p}$,

$$\delta = |\hat{p} - (1.1 * \hat{p})| \tag{5}$$

and for convenience sake we set $\hat{p_0} = \hat{p}$, and $\hat{p_1} = 1.1 * \hat{p}$. We can now compute n, the number of samples we need to take from the entire collection to validate that the results list contains at least the number of documents that are relevant.

Since

$$\delta = z_{1-\alpha/2}\sqrt{\frac{\hat{p_0}(1-\hat{p_0})}{n}} + z_{1-\beta}\sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n}}, \tag{6}$$

n is given by

$$n >= \left(\frac{z_{1-\alpha/2}\sqrt{\hat{p_0}(1-\hat{p_0})} + z_{1-\beta}\sqrt{\hat{p_1}(1-\hat{p_1})}}{\delta}\right). \tag{7}$$

Assuming a document collection of 550 million documents, and about 400 relevant documents[5], $\hat{p} = 7 \times 10^{-7}$. For a one-tailed 95% confidence level, $z_{1-\alpha/2} = 1.645$. For 10% confidence interval, $z_{1-\beta} = 1.282$, so

$$n >= \left(\frac{1.645 * \sqrt{7 \times 10^{-7}(1 - 7 \times 10^{-7})} + 1.282 * \sqrt{8 \times 10^{-7}(1 - 8 \times 10^{-7})}}{7 \times 10^{-8}}\right), \tag{8}$$

$$n > 35056. \tag{9}$$

In other words, tens of thousands of documents in the collection would need to be sampled.

Assuming this *was* possible, having determined that the result set contains at least the number of documents that are relevant, it is next necessary to randomly sample the results set to determine the proportion of it that is relevant. The same binomial equations can be applied. In this case the expected proportion of document that are relevant, $\hat{p}$ is near 1 (so we use 0.9), the confidence interval and confidence level might remain the same, so n is very small (about 7). From this the F_1 measure can be computed (i.e. we know f, r and f_r).

However, since such a large number of documents must be sampled to determine the number of relevant documents for a given query, this approach is infeasible.

The second approach, and an alternative to sampling the entire document collection, is to permute the results list and compute the precision (for example, p@10) of all possible orderings. In the case of 2000 results the number of permutations is $2000! = 6.4 \times 10^{868}$ which is too large to compute. However, with no recall component its not possible to know whether the recall base contains the best items (e.g. the lowest priced item). This is akin to known item finding where the known item is not known in advance and then measuring based on the assumption that the results list contains it. We do not believe this is valid way to measure quality.

The third approach, an approach used by Zobel [14] is to estimate the number of relevant documents in the collection using a number of different results lists for the same query. Each of a set of search engines is used to generate a results list for a given query. Then the first results list is examined and the number of relevant documents is noted. Then the second results is examined and the number of previously unseen relevant documents is noted, and so on for the third and other search engines. This is then plotted and extrapolated to the point at which a new search engine will not find any previously unseen relevant documents. Unfortunately, most search engines today work in essentially the same way (including BM25 ranking) and the diversity is insufficient to consider this to be a robust way of computing the number of relevant documents in the collection.

Each of the three ways we propose for computing the score for a single query's results list and irrespective of the results ordering are infeasible. We now turn our attention to the orderings a site provides rather than all possible orderings.

4.2 Buying: Offered Orderings

A more viable approach to measuring performance is to directly use the rank orderings offered by the site. In the case of Amazon, this would be the 6 orderings listed in Section 2, or the 10 orderings for Trademe. The obvious way is to compute the score for each list and to linearly combine and average.

[3] https://www.scrapehero.com/many-products-amazon-sell-january-2018/
[4] https://www.Trademe.co.nz/About-trade-me/Site-stats
[5] Roughly what we observe on Amazon today (mid 2018) for the query "iPhone X"

$$p = \sum_{a=1}^{|A|} \frac{\lambda_a p_a}{|A|} \tag{10}$$

where p is the precision and p_a is the precision score for ordering a of the A possible orderings, $|A|$ is the number of orderings, and λ_a is a weight for ordering a, and $\sum_{a=1}^{|A|} \lambda_a = 1$. If all rank orders are of equal importance,

$$\forall a, \lambda_a = \frac{1}{|A|}. \tag{11}$$

However, it is highly unlikely that all rank orderings are of equal importance to a site. On Trademe, "Best match" is the default, and "lowest price" appeals to bargain hunters, so we expect these to be weighted higher (more important) than "Title" or other orders.

One way to compute the λ_a weights is to compute the relative proportion of results lists presented in order a, others include the proportion of clickthroughs that come from the given list type, another is the proportion of sales from that list type, There are a multitude of possibilities, and most would require on-going observation as the proportions are likely to change based on the quality of the results, time, user location, and client device. In other words, there is a feedback loop.

The individual precisions, p_a, could be computed using any of the standard information retrieval metrics – that do not require an estimate of the recall. This might include P@n, Rank Biased Precision [10], or others. We note that P@3 has been used by some eCommerce sites as that is the number of results typically shown in the first page of results on a smart phone [7]. We also note that there is an implicit assumption in these metrics that the recall base is sufficiently large to contain the best answer for the given sort order – but the lowest priced item is the lowest priced item and it might not be in the recall base.

4.3 Browsing

A browsing user is interested in comparing the characteristics of multiple items. This might be obvious eCommerce features such as price, or delivery time, or it might be more esoteric such as whether a certain edition of a book is on the market.

We believe that a metric similar to Tolerance to Irrelevance, T2I [5], but for eCommerce is appropriate to measure browsing quality. That is, we envisage a user who continues to look down a results list until their tolerance to the irrelevant material is exceeded – we then ask how far down the result list the user is. This is similar to Cooper's Expected Search Length, ESL, of a simple ordering [4].

$$ESL = \sum_{i=1}^{k+\varepsilon} \overline{rel_i} \tag{12}$$

where k is the number of relevant documents we're looking for and ε is the maximum number of non-relevant documents we're prepared to tolerate (stopping after k relevant documents are found).

$\overline{rel_i}$ is 1 if the document at position i in the results list is not relevant, and 0 if it is relevant. ELS is the absolute number of irrelevant documents the user must view in order to see k relevant documents for a given query, which is then averaged over a number of queries. It also does not fall in the range [0..1].

We assume the user is interested in comparing k items, so we measure the effort required to find those k items. More precisely, we measure the inverse of that effort.

The effort to find one relevant document in one results list is simply the position of that item in the results list, $rank_1$. The inverse of which is the reciprocal rank for the query, RR, the mean over a number of queries, $|Q|$ is the mean reciprocal rank, MRR,

$$MRR = \frac{\sum_{i=1}^{|Q|} \frac{1}{rank_1}}{|Q|} \tag{13}$$

Generalizing this, to k relevant documents, RR_k,

$$RR_k = \frac{\sum_{i=1}^{k} \frac{i}{rank_i}}{k} \tag{14}$$

and the mean of this,

$$MRR_k = \frac{RR_k}{|Q|} \tag{15}$$

is the inverse of the effort the user must expend in order to observe k relevant documents. MRR_k is in the range [0..1] where 1 is best.

We observe that MRR_k is exactly equivalent to $MAP@k_r$ where k_r is the position in the results list of the kth relevant document (rather than the more usual kth position in the results list). An obvious extension is $MAP@k_{r\%}$

We also note the similarity to r-precision [1] where the precision is measured at position r in the results list where r is the number of relevant documents. Indeed, setting r to k_r on a query by query basis gives the precision at the point at which the user sees k relevant documents.

5 RELEVANCE

It is pertinent to ask what relevance means in the context of an eCommerce site. Goldberg et al. [7] suggest that for buying it might be defined by a book. That book encodes the difference between an individual user's expectation and the meaning of their query. They ask whether *basketball shoes* are a good answer to the query *basketball* or whether the user needs to be trained to ask for what they want as shopping is akin to known entity finding. Indeed, we accept that the definition of relevance for shopping is hard and requires further exploration as it is likely to include factors of price, seller rating, shipping time, and so on. However, a buy signal for a query is very strong evidence of relevance, and such signals might be mined from logs.

We believe that the definition of relevance for browsing is even more difficult to define – but it is clearly an item from an item set that the user wants to compare for some purpose. The purpose could be spelled out in a TREC-like topic definition. The set might be mined from user behaviour.

6 TASK PROPOSAL

We showed in Section 2 that both Amazon and Trademe support multiple rank orderings of the results sets, and in Section 3 that those rank orders are not of equal quality. In order to measure the quality of the site we proposed in Section 4.1 that it is not feasible to measure F_1 as the number of relevant documents cannot be known, and instead propose to measure a weighted average of the precision scores of each of the offered results orderings. In this section we provide more details on our proposed *task*. We propose to take a dump of a large-scale online eCommerce site such as Amazon or Trademe. This might be achieved either by agreement with the site, by crawling the site, or by extracting documents from a pre-existing crawl. There are several reasons such a site might choose to participate in such a dump. First, none of the data is proprietary, the data is already public facing and free. Second, providing a dump of the data to the research community is a marketing opportunity. Third, the results of research on such a document collection would be directly applicable by the group that makes data available, rather than requiring "porting" to a new document collection.

Acquiring a query log may be difficult as query data is proprietary, but a set of queries could be mined from a proxy log of a large institute that has not blocked eCommerce sites. The query is embedded in the URL of result page of both Amazon and Trademe, and extracting the query from that appears to be straightforward.

Values for λ_a could be estimated from a proxy log (although this might introduce bias). Both Amazon and Trademe embed the sort order in the URL. Either the proportions of queries using each sort order, or the proportion that lead to a buy, could be used.

Trademe and Amazon both support list and grid result presentation – and we believe that they should be measured differently. Set-wise evaluation appears, at the onset, to be a better metric for grids whereas rank-biased metrics appear to be better suited to lists. The quality of both presentation formats should be measured.

7 DISCUSSION

Both Trademe and Amazon support rank orderings that are direct inversions of each other. For example, the results list for "Highest price" should be directly computable from the results list for "Lowest price" by simply inverting the results list, but many not be because of tie breaks.

We believe that a well performing search engine that returns high quality documents irrespective of the rank order must be good at identifying relevant documents, and have both a low false positive rate and a low false negative rate. Hence, we believe that it will be a high accuracy search engine.

High accuracy recall identification is an interesting problem for many reasons. First, many years of assumptions about the ranking function pushing low quality results down the results lists no longer apply – the learning-to-rank pipelines in web search engines may not be applicable. Second, to be usable online, high accuracy with low latency is important. This raises new problems for IR efficiency research which generally uses algorithms such as WAND [2] or Anytime [9] which assume a pre-computed single rank ordering, and BitFunnel [8] many return too many false positives.

The similarity between some of the rank orderings (e.g. price low to high) and known entity search does not escape us. In the proposed task, however, the known entity is known to exist, but which document it is is not. Indeed, knowing whether or not any search engine has found the lowest priced relevant document does not appear to be easy. We only know that the lowest priced item amongst those assessed has been placed at the top of the list. The metrics we have proposed do not account for whether or not the most-relevant item is in the recall base. We leave for further work the development of metrics that account for this in absolute orderings. An obvious way to address this is to consider non-recalled but relevant documents as non-relevant documents. That is, if there are 3 relevant documents lower in price than the search engine returns then count that as 3 misses before the results returned by the search engine – however these might be weighted as a missing cheapest item is a greater mistake than a missing 25^{th} cheapest item.

8 CONCLUSIONS

In this short paper we examined two eCommerce sites and showed that they support different sort orders of the results list. We then showed that they are not equally good at ranking when using these sort orders and hypothesized that the problem is the quality of the recall set, those documents the search engine returns.

We suggested measuring the quality of the recall base irrespective of the presentation order and suggested that this as infeasible as it wasn't possible to known the number of relevant documents in the collection – and it wasn't possible to compute it.

We then proposed a weighted precision score as a metric and proposed methods of computing the weights – for buying. For browsing we developed a measure not dissimilar from tolerance to irrelevance, but based on MAP.

Finally we proposed the high accuracy recall task. For this task the search engine must identify as many relevant documents as it can without forfeiting precision – so that regardless of the presentation order the quality of the results is high.

We believe this is an interesting problem to tackle because it raises new questions about ranking, efficiency, and performance measurement. In future work we hope to build the collection and to launch the task.

REFERENCES

[1] J. A. Aslam, E. Yilmaz, and V. Pavlu. 2005. A Geometric Interpretation of R-precision and Its Correlation with Average Precision. In *SIGIR 2005*. 573–574.
[2] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien. 2003. Efficient Query Evaluation Using a Two-level Retrieval Process. In *CIKM 2003*. 426–434.
[3] K. Collins-Thompson, C. Macdonald, P. N. Bennett, F. Diaz, and E. M. Voorhees. 2014. TREC 2014 Web Track Overview. In *TREC 2014*.
[4] W. S. Cooper. 1968. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *Am. Doc.* 19, 1 (1968), 30–41.
[5] A. P. de Vries, G. Kazai, and M. Lalmas. 2004. Tolerance to Irrelevance: A User-effort Oriented Evaluation of Retrieval Systems Without Predefined Retrieval Unit. In *RIAO 2004*. 463–473.
[6] N. Fuhr. 2008. A Probability Ranking Principle for Interactive Information Retrieval. *IRJ* 11, 3 (2008), 251–265.
[7] D. Goldberg, A. Trotman, X. Wang, W. Min, and Z. Wan. 2017. Drawing Sound Conclusions from Noisy Judgments. In *WWW 2017*. 529–537.
[8] B. Goodwin, M. Hopcroft, D. Luu, A. Clemmer, M. Curmei, S. Elnikety, and Y. He. 2017. BitFunnel: Revisiting Signatures for Search. In *SIGIR 2017*. 605–614.
[9] J. Lin and A. Trotman. 2015. Anytime Ranking for Impact-Ordered Indexes. In *ICTIR 2015*. 301–304.
[10] A. Moffat and J. Zobel. 2008. Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM TOIS* 27, 1 (2008), 2:1–2:27.
[11] S. E. Robertson. 1997. Readings in Information Retrieval. Chapter The Probability Ranking Principle in IR, 281–286.
[12] M. Sharma, P. Sondhi, C. Zhai, and P. Kolari. 2018. A taxonomy of queries for e-commerce search. In *SIGIR 2018*.
[13] N. Sherlock and A. Trotman. 2011. Efficient sorting of search results by string attributes. In *ADCS 2011*.
[14] J. Zobel. 1998. How Reliable Are the Results of Large-scale Information Retrieval Experiments?. In *SIGIR 1998*. 307–314.