Quality metrics for search engine deterministic sort orders

Andrew Trotman[a,*], Vaughan Kitchen[a]

[a]University of Otago, Dunedin, New Zealand

ARTICLE INFO

Keywords:
eCommerce
evaluation
metrics

ABSTRACT

eCommerce search engines such as eBay and Amazon often allow the user to order their search results on deterministic features such as price, time, or distance from the seller. Using metrics such as precision at 10 documents (P@10), others have already shown that the quality of deterministically sorted results is lower than that of best-match (or relevance) sorted results, and that work is needed in order to improve result quality. But metrics such as P@10 are based purely on relevance, and do not reflect the order-feature: *cost* (be it price, time, or otherwise) – and it is hard to see how to improve a system without a metric that reflects the quality of the ordering. In this contribution we introduce a set of metrics that, using relevance *and* cost, measure the quality of deterministically sorted search engine results. We examine metrics from the perspective of the buyer, the seller, and the systems engineer. Using our new metrics (buying power (*bp*), buying power for K (*bp4k*), selling power (*sp*), and cheapest precision (P_c)) we re-evaluate the results of the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task" and demonstrate how to include cost in a metric designed to evaluate the quality of deterministically ordered lists of results.

1. Introduction

eCommerce sites such as eBay, Amazon, and Alibaba offer many thousands of items for their customers to choose from – so many that these sites are driven by search engines. Those search engines take a user's query and return a list of results for the user to choose from. Unlike web search, the user is then able to sort their results on a number of characteristics such as price (for example, low-to-high), or time (for example, auction-ending-soonest).

These *deterministic* sort orders have received less attention in the literature than *best-match* orderings. There are several reasons for this. Best-match algorithms such as BM25 (Robertson, Walker, Jones, Hancock-Beaulieu and Gatford, 1994) are near universal and so a large number of people have worked on them. Importantly, there are metrics such as *MAP* and *nDCG* that can be used to measure the performance of best match ranking in offline experiments (Järvelin and Kekäläinen, 2002). A frequent protocol is to conduct offline experiments to optimise against one of these metrics and then to conduct A/B experiments online to identify whether those improvements positively affect the users. Unfortunately, to the best of our knowledge, there are no such metrics for deterministic sort orders.

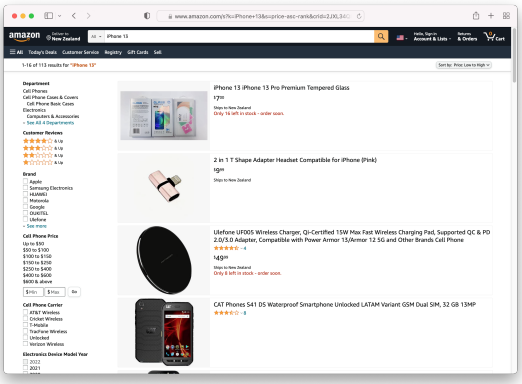
In this contribution we introduce three new classes of metrics for deterministically ordered results lists: buyer-centric, seller-centric, and system-centric. Using price-low-to-high as a running example, we develop separate metrics for each class. We use the queries, assessments, and runs from the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task" (Degenhardt, Kallumadi, Porwal and Trotman, 2019) to show that these new metrics, all of which include cost, correlate strongly with pre-existing metrics that do not – by introducing a fine-grained relevance-discriminator into our metrics we see a fine-grained (small) change when correlating with prior metrics.

Finally we suggest that these new metrics be used for deterministic orderings other than just price, including: distance (for point of interest search), time-starting-soonest (for job search), or even a combination features such as a price / time trade-off.

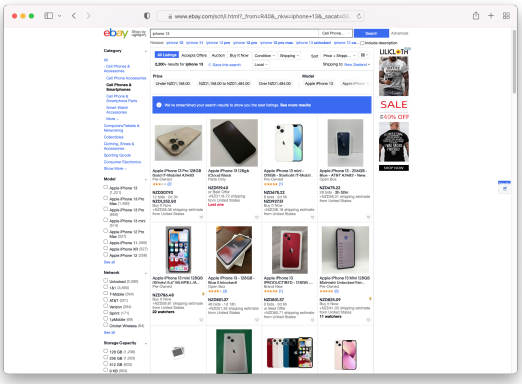
2. Problem definition

Our model of the ideal interaction between a user and an eCommerce search engine involves the user entering their needs into the search box, choosing "price-low-to-high" as the ordering, clicking "search", then examining the results list top-down until they find the required number of requested items. Our user does not want to pay more than necessary (they do not wish to be ripped-off), and they want what they searched for.

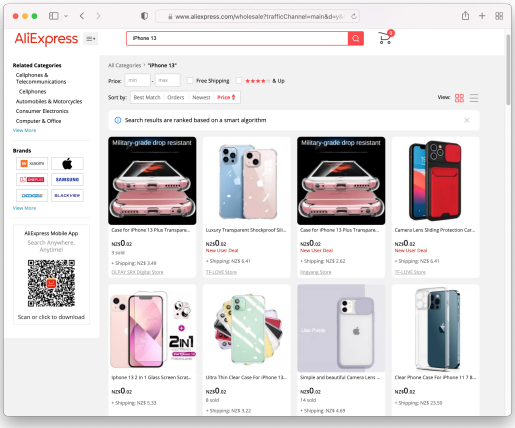
*Corresponding author
ORCID(s): 0000-0003-1253-7123 (A. Trotman)



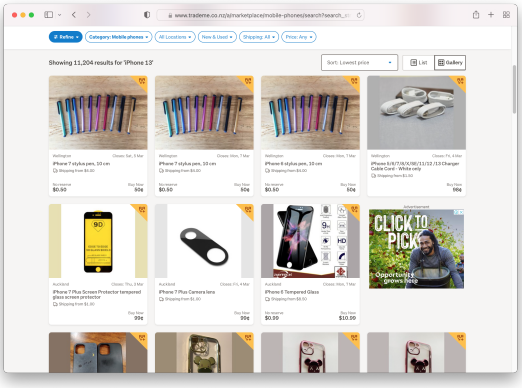
(a) Amazon.



(b) eBay.



(c) AliExpress.



(d) TradeMe.

Figure 1: The top results for the query "iphone 13" ordered price-low-to-high on 4 popular eCommerce sites. Of these sites, only one places an iPhone 13 in the initial results. The SERP for TradeMe has been scrolled to show the results grid. Figure best seen enlarged on screen.

For example, a user might search for "iPhone 13" ordered "price low-to-high". When doing so they expect to see a list of iPhone 13s, but unfortunately this is often not the case. To illustrate our point, on 1 March 2022 we did exactly this on Amazon, eBay, AliExpress, and TradeMe. The results are shown in Figure 1 where it can be seen that only one of these sites places an iPhone 13 in the initially displayed results. Although a single search is insufficient to robustly demonstrate the extent of this problem, it certainly is illustrative of a problem. We refer the reader to Spirin, Kuznetsov, Kiseleva, Spirin and Izhutov (2015) for a more extensive study.

Even if the results list only contains highly-relevant items, not all highly-relevant items are equal when sorted deterministically. For example, eBay orders on "Price + Shipping: lowest first", and so even if relevance does not change from user to user, the sort order might as it is dependant on the user's geographic location (shipping costs).

Additionally, a results list containing only highest-relevance items at some cut off (e.g. top 5) might not contain the lowest-priced highest-relevance item. A search for "iPhone 13", might find 3 different coloured 1TB iPhone 13 Pro Max at $2,999, and 2 different coloured 256GB iPhone 13 Pro at $1,999. But if the search engine did not find a 128GB iPhone 13 Mini (at $1,249), despite it being in the store, then it has failed to find a cheaper item and place it at the top of the results list – and this should be reflected in the quality metrics. Such a failure is not reflected in relevance-alone metrics such as *MAP* and *nDCG*. A combination of relevance and cost is needed.

It is easy to simply state that there is only one possible relevant item per price-ordered query (thus turning eCommerce search into known-item search), but this is also insufficient. The iPhone 13 comes in many colours, and

so there are several possible cheapest items. The query might be ill-specified (there are several models of iPhone 13) and so it is not certain what, exactly, the user wants. Additionally the search engine should be trained to identify the set of possible relevant items in case the retailer sells out of the one-possible-item but has alternatives.

In an auction setting, where the sellers write the listings, keyword matching between a query and an item might fail due to the vocabulary miss-match problem. A listing for an "Apple Phone Model Thirteen" is most likely to be an iPhone 13, despite not containing the word iPhone or the number 13. Trotman, Degenhardt and Kallumadi (2017) outline how eBay perform query-rewrite to address this problem. Goldberg, Trotman, Wang, Min and Wan (2018) discuss how eBay define relevance. In short, there must be a notion of relevance (an item the user might buy given the query), but to measure the quality of a price-low-to-high ordered results list, relevance and cost must both be used.

Some eCommerce sites allow the user to rank the results list in order of time. eBay, for example, offers "Time:ending soonest". Again, relevance alone is insufficient to determine the quality of the results list – which must be measured with a combination of relevance and time. Our user might also have a preferred time / price trade-off. In this case the search engine should include the ability to sort on this, and the quality metric should measure the quality of the results list relative to the trade off.

The metrics we introduce use not only relevance, but also cost. Without loss of generality we describe our metrics as if it were price. But it might be a time cost when looking for a job, or a distance cost for visiting a point of interest. Indeed, it might be a combination of atomic costs, such as distance with fuel consumption, or price with shipping time.

In our experiments we make a number of assumptions. First, we assume the assessments are sound (i.e. correct) and complete (i.e. there are no missing assessments). Following the example of TREC, we assume any unassessed documents are non-relevant. We do not examine the problem of obtaining relevance assessments in an eCommerce environment – that is left for future work. Second, we assume that the granularity of the assessments is correct. By this we mean that if there are substitutes for a product, this is correctly identified in the assessments. For example, for the query "Kellogg's Cornflakes", the assessor might mark supermarket-brand Cornflakes and "Kellogg's Frosted Flakes" as relevant. Third, we assume an eCommerce site is selling items (or experiences) and these are represented in a search engine as documents. Consequently, in the remainder of this paper we use the terms "document" and "item" interchangeably. Fourth, we assume binary relevance. That is, either a user would or would not purchase the item if it was the only item recommended to them. Fifth, the document collection contains at least one known-relevant document for each query being evaluated (which is conventional in off-line IR evaluation experiments).

3. Related work

3.1. Algorithms for deterministic ordering

Spirin et al. (2015) provide evidence of the low quality of deterministic sort orderings on eCommerce and job-search sites. They issued 25 queries to 10 sites, applied deterministic sorting, and measured the quality of the results using P@10 (and other metrics). They found that "the average Precision@1 is 0.44, Precision@5 is 0.45, and 61% of queries have the Precision@10 below 0.5". For "best match" search they measured P@10 of 0.86. Trotman, Kallumadi and Degenhardt (2018) use a single head query on Amazon and TradeMe, but with many different sort orders. Figure 1 shows our single query demonstration of the continued persistence of a problem.

Several groups of researchers have examined algorithms to turn a relevance ranked list of results into a deterministically ordered list of results. Spirin et al. (2015) introduce a dynamic programming algorithm that produces the optimal result. They use date as the order and they optimise against *DCG*.

Nardini, Trani and Venturini (2019) observe the high computational cost of computing the optimal list using the algorithm of Spirin et al. (2015). They introduce ε-Filtering, a fast approximately-optimal algorithm for computing high relevance deterministic orderings. They use the GoogleLocalRec data set and sort on distance for some experiments and the AmazonRel data set price sorted as another. They optimise against *DCG* and *DCG-LZ*.

Unfortunately *DCG* uses relevance alone and is cost-agnostic. When measuring a results list at low cut-off (e.g. the first page of 10 results), the results list might be exclusively highest-relevance items, but *DCG* does not penalise the results list for missing the lowest-cost most-relevant items – and so the metric could easily be giving a false indication of quality.

3.2. IR metrics

There are dozens of metrics for information retrieval. In Section 7 we compare against F_1, *AP*, *RR*, *ESL*, *RBP*, and *l2h_ndcg*, and so we show the derivation of those metrics in this section. The notation we use for describing these

metrics is in common use, but for the metrics we introduce we prefer a sequence based notation as both the results list and the assessments are ordered lists. This section also includes a discussion on prior work that used cost or effort in the relevance metrics, metrics we believe are better suited to environments other than eCommerce.

Metrics vary mostly in the user model. Precision, for example, assumes a *set* of results and measures the proportion of those documents that are relevant ($P = \frac{|found\_relevant|}{|found|}$). Precision says nothing about the quality of those results other than that they are relevant.

Average Precision, (*AP*, or *MAP* when averaged over a set of queries), is used for *lists* of results, and is defined as

$$AP = \frac{\sum_{i=1}^{n}(P_i \times r_i)}{\sum_{j=1}^{d} r_j}, \tag{1}$$

where P_i is the precision at rank i, d is the number of documents in the collection, $\sum_{j=1}^{d} r_j$ is the number of known-relevant documents for the query, n is the length the results list, and

$$r_i = \begin{cases} 1, & \text{if the document at position } i \text{ is relevant,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Average precision assumes a user looking down a results list from top to bottom. The score accumulates $\frac{P_i}{\sum_{j=1}^{d} r_j}$ at each point at which the user encounters a relevant document.

Discounted Cumulative Gain (*DCG*) (Järvelin and Kekäläinen, 2002) moves away from the binary definition of relevance, and in doing so models a user who believes some documents are more relevant than others. It assumes a results list and a user examining that list top to bottom. Cumulative Gain, *CG*, is defined as

$$CG = \sum_{i=1}^{n} r_i, \tag{3}$$

where r_i is the gain associated with the document at position i in the results list (often computed as a score in the range [0,1]). Cumulative gain, much like average precision, is accumulating score as the user reads down the results list and finds more relevant material. It assumes all positions in the results list are equally important. As it is simply accumulating, it might not result in scores in the range [0,1].

Discounted cumulative gain, *DCG*, discounts the amount of gain by dividing by some function of how far down the results list the user has explored,

$$DCG = \sum_{i=1}^{n} \frac{r_i}{\log_2(i+1)}. \tag{4}$$

DCG also accumulates gain. Problematically, the maximum scores possible with *DCG* can vary from query to query making it hard to compare the performance on one query with the performance on another – such a situation occurs when one query has only highly relevant documents (that might score a 1) but another only has marginally relevant documents (that might score a 0.3).

To alleviate this, a normalised version, *nDCG* (Järvelin and Kekäläinen, 2002), can be computed as the ratio of the *DCG* score to that of the maximum possible score for that query. The *DCG* of the results vector divided by the *DCG* of the ideal results vector (*IDCG*).

Metrics such as rank biased precision (*RBP*) (Moffat and Zobel, 2008) assume a user starts at the top of the results list and reads down, each time deciding whether or not to continue, which they do with some fixed probability p,

$$RBP = (1-p) \times \sum_{i=1}^{n} \left(r_i \times p^{i-1}\right). \tag{5}$$

Amongst other probabilities, Moffat and Zobel (2008) use a continuation probability of 0.95, representing a persistent user who has a 60% likelihood of going past 10 results and a 35% likelihood of going past 20 results. *RBP*, like *nDCG*, assumes a graded relevance score for r_i.

Many eCommerce queries are for specific items. Those queries can be thought of as known-item searches. The metric used for this kind of search asks where, in the results list, that known item can be found. The user model assumes that more work (looking further down the results list) is bad, and so a lower score is given the further down the list the known item is found. The quintessential metric for known-item search is Reciprocal Rank (Bhargav, Sidiropoulos and Kanoulas, 2022), *RR*, or *MRR* when averaged over several queries,

$$RR = \begin{cases} \sum_{i=1}^{n} \frac{1}{i \times r_i}, & \text{if there is 1 found and relevant,} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Since there is only one possible answer for known-item search, r_i is true at most once giving $\frac{1}{i}$ where i is the position of the relevant item.

If K such items are wanted then the mean rank of the K items might be used rather than the position of the first item, but if there are fewer than K relevant items in the results list then $RR_k = 0$, giving

$$RR_k = \begin{cases} \frac{1}{K} \times \sum_{i=1}^{v} \frac{1}{i \times r_i}, & \text{if there are} \geq K \text{ found and relevant,} \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where v is the depth at which the user finds the K^{th} relevant item. Bailey, Moffat, Scholer and Thomas (2015) use the less fine-grained averaging of $\frac{K}{\mathbb{K}}$ where $\mathbb{K}$ is the rank of the K^{th} relevant item. The difference is analogous to micro versus macro averaging.

In *ESL*, Cooper (1968) counts the number of non relevant documents before the first relevant document, giving a score that goes up as the quality of the system goes down,

$$ESL = \begin{cases} \sum_{i=1}^{n} \big((i-1) \times r_i\big), & \text{if there are} \geq 1 \text{ found and relevant,} \\ \infty, & \text{otherwise.} \end{cases} \tag{8}$$

Smucker and Clarke (2012) introduce a discount on the information gain as a function of time spent in their time-biased gain metric, *TBG*. Prior metrics assume a user spends a constant time per document on the results list, but this is unrealistic as some can be discarded as non-relevant without examination (or from examining just the snippet) and users spend more time on more-relevant documents. *TBG* discounts gain as a function of time spent at each point in the results list – including for non-relevant documents. Using it for eCommerce would require re-calibration, and we are interested in more than just time.

Jiang and Allan (2017) introduce a cost for different relevance grades (including non-relevant) in their adaptive effort metrics. That cost might be measured from user behaviour logs, or predefined. Our cost model is similar to their effort vector, but is not constant per non-relevant document and can partially accumulate even for relevant documents. Our contribution can be thought of as metrics for a cost-enhanced relevance model.

Zhang, Liu, Li, Zhang, Xu and Ma (2017) introduce the Bejeweled Player Model, *BPM*, which accumulates *benefit* for a *cost*, stopping when either passes a threshold. Their model is abstract and, no-doubt, eCommerce metrics could be described using their model by, for example, setting the maximum cost to the the user's budget (financial or otherwise) and the maximum benefit to the user's desired purchase quantity.

We do not explore the literature on information foraging (Azzopardi, Thomas and Craswell, 2018) as we wish to measure the quality of a results list rather than the results page (that includes headers, footers, right rails, etc.). We also do not explore economic models of search (Azzopardi, 2014) as they are used to model users rather than to measure the quality of a results list.

3.3. IR metrics for eCommerce

Trotman et al. (2018) suggest that if the sort-order is not known in advance, as might happen at the first stage of a ranking pipeline, then the best that a search engine can do is to return as many highly relevant documents as possible. They claim that the quality of such a set is best measured using F_1 of precision and recall ($R = \frac{|found\_relevant|}{|relevant|}$).

F_1 is defined as

$$F_1 = \frac{2 \times P \times R}{P + R}. \tag{9}$$

But, since eCommerce collections can contain tens of millions of documents, the number of relevant documents cannot be known so Trotman et al. (2018) propose three methods to estimate this.

Trotman et al. (2018) also suggest that if the quality of a query is being measured then a weighted average of the site's possible sort-orderings might be used. That is, if B is the set of sort orderings seen for a given query, with relative frequencies λ_b, and the precision for each ordering, P_b, is known then the precision of the query, P_q, is

$$P_q = \sum_{b=1}^{|B|} \frac{\lambda_b \times P_b}{|B|}. \tag{10}$$

The metric used for the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task" was *l2h_ndcg*. This metric has never been described and so we reverse engineered it from the evaluation tool.[1] The metric is a binned version of *nDCG*. The bins are exponentially increasing price groups and are used in place of relevance in *nDCG*. The bin for a given item, B_c is computed from the item price, c, the lowest-cost relevant item, C, and the highest-cost relevant item, H thus,

$$B_c = b + 1 - \left\lfloor \ln\left(1 - \frac{(c - C) \times (1 - e)}{(H - C) \times \frac{1-e}{1-e^b}}\right)\right\rfloor, \tag{11}$$

where b is one fewer than the number of bins ($b = 5$, meaning 6 bins, was used in the challenge and in our experiments). B_c is at its maximum when the price of the item is lowest and at its minimum when the price of the item is at its highest (i.e. $B_c = 1$ when $c = H$, and $B_c = b + 1$ when $c = C$). Non-relevant documents score 0.

The bins are then used as the relevancy score for *l2h_DCG*,

$$l2h\_DCG = \sum_{i=1}^{n} \frac{B_{c_i}}{\log_2(i + 1)}, \tag{12}$$

where *l2h_ndcg* is computed from *l2h_DCG* and the ideal gain vector *l2h_IDCG*.

3.4. Accumulating loss

Most metrics for information retrieval measure some form of accumulation of gain. The more relevant material the user sees the better the score. *ESL*, and to some extent *TBG*, are exceptions, instead accumulating loss of time scanning a results list. We believe that a quality metric for eCommerce search should reflect over-spending. That is, it should accumulate loss.

The next sections introduce metrics that reflect how much loss the user made by comparison to the minimum possible loss. Our metrics are all in the range [0,1], and larger is better. We discuss the metrics on a query by query basis, but assume they will be averaged over a number of queries. We assume the arithmetic mean, but note that other averaging such as Pq from Equation (10) might be used. We divide our metrics into three groups: buyer-centric metrics, seller-centric metrics, and system-centric metrics.

4. Buyer-centric metrics

In this section we describe metrics that measure the quality of a site with respect to a user trying to buy the cheapest instances of a given item (where cheap may mean time, price, distance, or any other cost-based measure). We first examine the purchasing of a single instance of an item, then multiple instances of that item. If the user is purchasing a set of items (for example a jar of jam and a box of cornflakes) then we treat each purchasing item in that set as a separate query and average over the queries.

The symbols used to describe the metrics are defined as they are introduced, but Table 1 can be used as a reference.

4.1. Buying power

We illustrate with a user entering "cornflakes" into the search box, selecting "price-low-to-high", and examining the top result in the results list. That result could be: The cheapest relevant item, a relevant item that is not the cheapest, or a non-relevant item.

[1] https://github.com/eBay/sigir-2019-ecom-challenge

Table 1
Symbols used in the metrics introduced in this paper. We assume a single query with a single results list is being scored and so for simplicity, q subscripts are dropped.

Symbol	Meaning
i, j, s	An index into a sequence.
A	The sequence of all *assessed-relevant* documents in the collection, ordered from lowest cost to highest cost.
$\|A\|$	The number of known relevant documents (the size of A).
A_j	The j^{th} element in A, A_1 being the lowest-cost relevant item in the collection.
A_j^c	Cost of A_j.
A_j^r	The binary relevance of A_j (1 for true, 0 for false).
L	The sequence of found documents in the results *list*, ordered from 1 (the start of the results list).
$\|L\|$	The number of found documents (the size of L).
L_j	The j^{th} element in L, L_1 being the first result in the results list.
L_j^c	The cost of L_j.
L_j^r	The binary relevance of L_j (1 for true, 0 for false).
D	The depth of evaluation (e.g. $D = 10$ in P@10).
K	The number of relevant items the user wishes to purchase.
I	The index of the K^{th} relevant item in L.
m	The number of results the user examines. As the user stops shopping when K relevant items have been found, m is not the same as the more usual cut-off point, n seen in previous metrics. $m = \min(I, \|L\|, D) : \sum_{i=1}^{I} L_i^r = K$.

In the first two cases where the top item is relevant, it is pertinent to ask how much value for money the user will get if they purchase that item. In other words, how much might the user spend by comparison to the cost of the cheapest relevant item the retailer can supply, or,

$$bp_r = \frac{A_1^c}{L_1^c}, \tag{13}$$

where bp_r is the *buying power* of the user's spending *if the item is relevant*, A is the sequence of assessed-relevant items ordered low to high on cost, A_1^c is the cost of the lowest-cost relevant item the retailer can supply, L is the sequence of results in the results list, and L_1^c is the cost of the item at position 1 in the results list. If this user buys the lowest-cost relevant item on the site then $bp_r = 1$. If the user buys a relevant but over priced item then bp_r represents the proportion of the user spending that is value, and $1 - bp_r$ is the over-spending proportion. For example, if our user pays \$3.90 for a box of Kellogg's Cornflakes but could have paid \$3.30 for a bag of Sanitarium Cornflakes, then only $bp_r = 3.30/3.90 = 0.84$ of their spending was value, whereas $1 - bp_r$ (0.16, or 16%) of their spending was wasted – assuming the items are interchangeably-relevant. With a constant minimum cost, A_1^c, as the amount spent, L_1^c, goes up, the bp_r goes down. This is similar to computing the difference between the ideal vector and the results vector in *nDCG*. Conversely as L_1^c tends to A_1^c the score tends to 1. With a relevance-only metric such as *RR* it is not possible to distinguish been the lowest-cost relevant item, and any other relevant item because under *RR* any relevant item scores 1.

If the item the user sees is not relevant (for example: Countdown Tuna Flaked Lemon & Cracked Pepper[2]) then there would be no value in purchasing that item, so

$$bp_{\bar{r}} = 0, \tag{14}$$

where $bp_{\bar{r}}$ is the buying power of the user *if the item is not relevant*. Described this way, bp_r and $bp_{\bar{r}}$ give a fine-grained document relevance score computed from relevance, ideal price, and spending. These might be plugged into any prior graded-relevance metric such as *nDCG*, but doing so ignores user behaviour.

Our disgruntled user might now choose to examine the next result in the results list. This process is repeated until either a relevant item is found, the supplier can no longer supply items, or the user gives up. Each time the user is

[2]On 16 May 2022 this was the first result on the Countdown Supermarket web site when searching for cornflakes and sorting the results "Price Low to High". This is quite possibly due to the word "flaked".

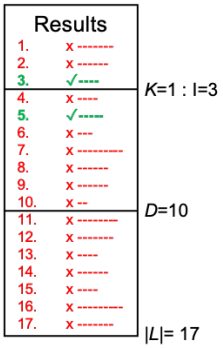


Figure 2: We model a user who examines the results list top to bottom and who stops after seeing K relevant items; or gives up sooner if they have seen D items; or reaches the bottom of the results list of $|L|$ items. Relevant items are shown in green with a tick, non-relevant items are shown in red with a cross.

presented with a non-relevant item the system receives a metric's penalty equivalent to the cost of the non-relevant item. That is, buying power, *bp*, accumulates loss as if the user had purchased (and not returned) all results in the results list up to and including the first relevant item. This decision is not arbitrary, imagine a user interacting with a digital assistant and saying "send me your cheapest box of cornflakes" and being sent "Countdown Tuna Flaked Lemon & Cracked Pepper". The postage price of returning the tuna is higher than the purchase price of the tuna so our customer does not return it, they simply return to their digital assistant and say "that isn't cornflakes, send me your cheapest box of cornflakes". Each time this happens they accumulate a loss equal to the cost of the item that was shipped. In other words, the cost of the first relevant item is the cost of all the items in the results list up-to and including the cost of the first relevant item.

More succinctly, if the retailer does not have the item then the buying power is 0, otherwise it is the ratio of the cheapest relevant item to the sum of the costs of each item up-to and including the first relevant item,

$$bp = \begin{cases} \frac{A_1^c}{\sum_{i=1}^{m} L_i^c}, & \text{if there is a relevant item at or before } m, \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where L_i^c is the cost of the item at position i in the results list and where m is the maximum depth the user is prepared to search.

Our user model accounts for the stopping criteria, m, being the minimum of three possible values. It could be the first relevant item in the found sequence, in which case $m = I, I : \sum_{i=1}^{I} L_i^r = K, K = 1$, where L_i^r is the binary relevance of item i in the sequence (1 for relevant and 0 for not relevant). Or m might be the length of the results list ($|L|$). Finally, m might be the point at which the user gives up, D (the maximum depth of evaluation). These three criteria are illustrated in Figure 2, but formally and more generally,

$$m = \min(I, |L|, D) : \sum_{i=1}^{I} L_i^r = K. \tag{16}$$

The difference between our user model and the *nDCG* user model is important to note. Our user finds what they want and stops, doesn't find what they want because the store doesn't have it, or gives up looking. The *nDCG* user keeps going until some fixed cut-off in the results list, regardless of whether or not their information need has been fulfilled.

Two hypothetical examples of buying power are given in Table 2. Our user is interested in purchasing 1 relevant item. The lowest-cost relevant item has a cost of $A_1^c = \$2.50$. In the example on the left, the first relevant item, at position $m = 3$ has a price of \$5, and the sum of costs up-to and including that item, $\sum_{i=1}^{m} L_i^c$, is $\$1 + \$2 + \$5 = \$8$. The buying power is, therefore, $2.5/8 = 0.3125$. In the example on the right the buying power is 0.4545. A metric such

Table 2
Example calculation of buying power and buying power for K using a hypothetical results list showing costs and relevance decisions. There are 3 relevant items for this query. The lowest-cost relevant item, with a cost of \$2.50, was not found on the left, but was on the right. The buying power for 1 relevant item, *bp* is computed as the ratio of the cost of the lowest-priced relevant item (\$2.50) to the sum of the costs up-to and including the first relevant item. On the left this is $2.5/8 = 0.3125$. The buying power for $K = 2$ items, $bp4k_{K=2}$, is computed as the ratio of the sum of the lowest-cost K relevant items ($\$2.50 + \$5$) to the price of all items the user sees up-to and including the K^{th} relevant item, on the left: $7.50/28 = 0.2679$.

Relevant	Price (L_i^c)	$\sum_{i=1}^{m} L_i^c$	*bp*	$bp4k_{K=2}$	Price	$\sum_{i=1}^{m} L_i^c$	*bp*	$bp4k_{K=2}$
No	\$1.00	\$1.00	0	0	\$1.00	\$1.00	0	0
No	\$2.00	\$3.00	0	0	\$2.00	\$3.00	0	0
Yes	\$5.00	\$8.00	0.3125	0	\$2.50	\$5.50	0.4545	0
No	\$9.00	\$17.00	-	0	\$9.00	\$14.50	-	0
Yes	\$11.00	\$28.00	-	0.2679	\$11.00	\$25.50	-	0.2941
No	\$12.00	\$40.00	-	-	\$12.00	\$37.50	-	-

as *RR* is unable to distinguish between these examples because, in both cases, the first relevant result is at position 3 in the results list and so $RR = 0.3\dot{3}$. As with *RR*, if no relevant items are found then the score is 0.

The buying power metric assumes the user is interested in only one relevant item. If that item is known to exist then this is known item search. We next ask what happens if the user is interested in more than one item.

4.2. Buying power for K

A user ordering their weekly groceries online might want more than one of a given item. For example, they might want two boxes of cornflakes. Extending buying power to K instances of an item is straightforward.

Assuming each item is a separate listing in the results list (but this need not be the case), which might be the case for items on an auction site, then we ask: what is the ratio of the minimum possible spending to the amount the user would spend in order to purchase everything on the results list up-to and including K relevant items,

$$bp4k = \begin{cases} \dfrac{\sum_{j=1}^{K} A_j^c}{\sum_{i=1}^{m} L_i^c}, & \text{if there are} \geq K \text{ relevant at or before } m, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

If there are fewer than K relevant items in the results list then the search engine cannot fulfill the user's needs – which is analogous to not finding a known item in known item search. We, consequently, give a score of 0 (even if the search engine can fulfill a request for $K - 1$ items). If the search engine puts the K lowest-priced items in the top K rows of the results list then the score is 1. As the relevant items are pushed further down the results list the score tends to 0 because the denominator increases but the numerator is constant.

Two hypothetical examples of buying power for K are shown in Table 2. There are 3 relevant items priced at \$2.50, \$5.00, and \$11.00. Our user wishes to purchase $K = 2$ items, so the minimum possible spend at the store, $\sum_{j=1}^{K} A_j^c$, is $\$2.50 + \$5.00 = \$7.50$. While scanning top-to-bottom for $K = 2$ relevant items the user sees $m = 5$ items. On the left the total cost, $\sum_{i=1}^{m} L_i^c$ is $\$1.00 + \$2.00 + \$5.00 + \$9.00 + \$11.00 = \$28.00$. So , $bp4k = 7.50/28.00 = 0.2679$. In the example on the right, $bp4k = 7.50/25.50 = 0.2941$. Using average precision, the two cases are indistinguishable, being $(\frac{1}{3} + \frac{2}{5})/3 = 0.2444$, and note the final division by the number of known relevant items even though the user is only looking for $K = 2$ of them. Had there been fewer than 2 relevant items in the results list then *bp4k* for this query would be 0 as the search engine would have failed to fulfill the user's need.

On some sites each result in the results list can represent multiple instances of the same item (Amazon, for example, often lists the number of items in stock such as "Only 2 left in stock", and eBay has multi-SKU (stock keeping unit)). We model a user examining the results one after another. If an item is not relevant then it is not relevant for that group of items (or "blue link"). If it is relevant then multiple items can be supplied without further penalty. That is, the penalty is only given once per "blue link" but the reward is given for each unit the "blue link" can deliver.

We next ask whether the seller has a good display of relevant items – that is, how good is the shop-front?

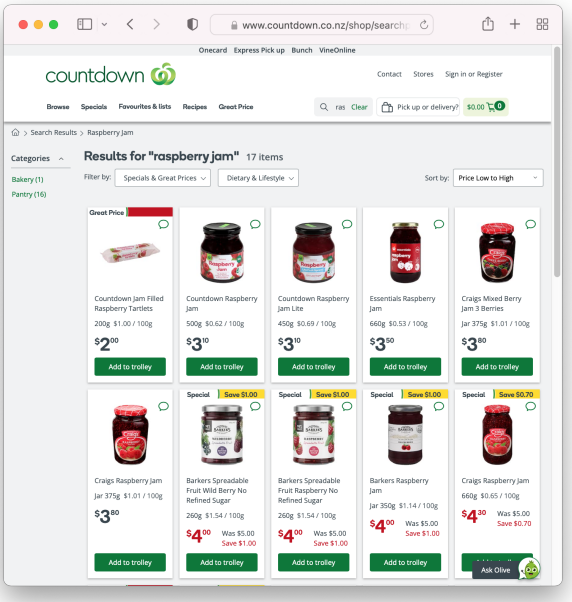


Figure 3: Top 10 results for the query "raspberry jam" ordered price-low-to-high on a supermarket website showing multiple items at the same price as well as non-jam results.

5. Seller-centric metric (selling power)

We now shift our attention from that of the buyer to that of the seller. Iyengar and Lepper (2000) conducted a study on consumer choice. They set up a jam tasting stall at a Menlo Park supermarket over two consecutive Saturdays and, hourly, swapped the number of varieties on display from 6 to 24. Consumers who partook of the tasting received a discount voucher surreptitiously identifying how many jams they were exposed to. They found that more customers stopped when there was greater choice, but more purchased when there was less choice. The jam manufacturer made 28 varieties, so in both cases it was necessary to choose the best items for display given a fixed number of "slots".

Kelly and Azzopardi (2015) conducted the equivalent study on search engine result page (SERP) length. They examined the cases of 3, 6, and 10 blue links per page and found that with their 36 users there were significant interaction differences – albeit a pattern was not obvious. Goldberg et al. (2018) suggest that eBay value precision at a low depth (specifically, P@3) because that is the number of results that appear on a typical smartphone screen.

We ask: if there are a limited number of slots available to fill then how effective is the search engine at filling those slots?

As an example of this problem, on 9th May 2022 we conducted a search for "raspberry jam" on the Countdown Supermarket web site and ordered the results price-low-to-high. The top 10 results are shown in Figure 3, where it can be seen that the first result is not jam but contains jam, that 2 more of the top 10 are jam other than raspberry (albeit containing raspberries), and that there are several items of the same price (low sugar and regular sugar varieties). Seventeen items were found, 12 of which were raspberry jam. With this query our supermarket cannot know whether the shopper is looking for low-cost raspberry jam, low-sugar raspberry jam, or "spreadable fruit", and so there is more than one possible answer (this is not known-item search). The task is to fill the results list with varieties of raspberry jam, ordered price-low-to-high, and we wish to quantitatively measure the slot-filling performance so that we can determine the effect of any changes to the algorithm that selects which items to show.

We assume that a cheap and relevant item is a better slot-filler than an expensive and relevant item or a non-relevant item. For the base case of the single slot at the top of the results list, the selling power, *sp*, is the same as the buying

Table 3
Example calculation of selling power. In this example there are 4 relevant items with prices \$1, \$2, \$3, and \$4. The search engine places a relevant item costing \$2 in the first "slot", gaining a selling power for that slot of $\frac{1}{2}$ because it could have placed an item costing \$1, but instead placed an item costing \$2. The item at the second slot is not relevant, gaining a selling power of 0. For the third slot, the search engine might have put the second lowest cost item at \$2, but instead put a relevant item that cost \$4, gaining a selling power of $\frac{2}{4}$. In total there are 3 slots and so the final selling power score is $\frac{1}{3} \times \left(\frac{1}{2} + 0 + \frac{2}{4}\right) = 0.3\dot{3}$.

Relevant	Price	Slot Power	Sum of Slot Powers
Yes	\$2	$\frac{1}{2}$	0.5
No	\$3	0	0.5
Yes	\$4	$\frac{2}{4}$	1
Selling Power			$0.3\dot{3}$

power,

$$sp = \begin{cases} \frac{A_1^c}{L_1^c}, & \text{if the result is relevant} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

where A_1^c is the lowest-cost relevant item and L_1^c is the cost of the item in that slot. If the lowest-cost item is found then $sp = 1$, if the found item is non-relevant then $sp = 0$, otherwise it is the effectiveness of the slot at showing low cost relevant items (the value the buyer achieves by buying from that slot).

We assume that the perfect store with $|L|$ slots to fill will list the $|L|$ lowest-cost relevant items in those slots. That is, we do not account for practices such as deliberately listing expensive items in order to entice the user to purchase more profitable items. We also do not account for query ambiguity and consequently result list diversity, instead leaving that for future work. As the results list sort order is deterministic, those items must be listed from lowest cost to highest cost.

With these assumptions we can compute the ideal item ordering in much the same way that *nDCG* computes the ideal gain vector. Whereas the ideal gain vector in *nDCG* is in decreasing order of relevance, the ideal vector for selling power is increasing in cost.

At each slot we can now compute the selling power of that slot. Although there are $|L|$ slots, there may be $|A| < |L|$ relevant items in the collection, so we limit the evaluation to the minimum of the two, $\min(|A|, |L|)$,

$$sp = \frac{1}{\min(|A|, |L|)} \sum_{s=1}^{\min(|A|,|L|)} \begin{cases} \frac{A_{\sum_{i=1}^{s} L_i^r}^c}{L_s^c}, & \text{if } L_s^r \text{ is 1 (i.e. } L_s \text{ is relevant),} \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

where L_i^r is the binary relevance of the item in slot i, and so $\sum_{i=1}^{s} L_i^r$ is the number of relevant items that appear in the results list up-to and including slot s. Ignoring slots filled with non-relevant items, the cost of the lowest-cost item that might have been put in slot s is $A_{\sum_{i=1}^{s} L_i^r}^c$.

Ignoring non-relevant slots this way might appear arbitrary, however it is not. If the search engine fails to place the lowest-cost as-yet unseen item in a slot then it is given another opportunity to place that item in the next slot – not a reprieve from finding that item. Finally, the scores for each slot are summed and the mean over the total number of slots is computed.

As an example, assume there are 3 slots and 4 known relevant items (at costs: \$1, \$2, \$3, and \$4). Given a results list with 1 relevant item at a cost of \$2, 1 non-relevant item at a cost of \$3, then 1 relevant item at a cost of \$4. Selling power is computed as $\frac{1}{3}\left(\frac{1}{2} + 0 + \frac{2}{4}\right) = 0.3\dot{3}$. This is further illustrated in Table 3.

Next we examine metrics for measuring the quality of early stages in the ranking pipeline.

Table 4
Example calculation of cheapest precision. In this example there are 4 known-relevant items priced at \$1, \$2, \$3, and \$4. The search engine returns 2 items for the query, and so the relevant set for computing cheapest precision consists only of the 2 lowest-cost relevant items (\$1, and \$2). Three different possible results sets are shown. On the left, 1 result from the lowest-cost relevant set is identified in the 2 results and so the score is 0.5. In the middle, despite finding 2 relevant items, those items are not in the set of the 2 lowest-cost relevant items and so the score is 0. On the right 1 relevant lowest-cost item and 1 relevant but not lowest-cost item are found resulting is a score of 0.5 as the relevant but not lowest-cost item is not a good candidate to pass up the ranking pipeline. Scored with set-based (not cost-based) precision, the middle and right examples are indistinguishable.

Relevant	Relevant Lowest-Cost	Price	Relevant	Relevant Lowest-Cost	Price	Relevant	Relevant Lowest-Cost	Price
Yes	Yes	\$1	Yes	No	\$3	Yes	Yes	\$2
No	No	\$9	Yes	No	\$4	Yes	No	\$3
P_c@4		0.5	P_c@4		0	P_c@4		0.5
P@4		0.5	P@4		1	P@4		1

6. System-centric metrics (cheapest precision)

Both buying power and selling power implicitly assume the user is only going to examine a small number of items in the results list (for example, the top 10). When developing an eCommerce system with a multi-stage ranking pipeline it is useful to measure the performance of each stage in the pipeline. In this section we introduce a metric for this purpose.

Set-based precision cannot be used because it only expresses the proportion of the results that are relevant. It says nothing about whether or not those items will be "good" items later in the pipeline (i.e. should be in the top-k results when sorted on cost).

We introduce a subtle, but important, change to set-based precision for exactly this purpose. We simply compute the precision of the results set on the lowest-cost known-relevant items – the items that we want to place in the top-k at the end of the pipeline.

Assuming the given stage of the pipeline has found and returned $|L| \leq D$ items to the next stage, we sort the assessments on cost and use the $\min(|A|, |L|)$ lowest-cost relevant items in the collection to calculate the cheapest precision, P_c:

$$P_c = \frac{\left| L \bigcap \{A_j\}_{j=1}^{\min(|A|,|L|)} \right|}{|L|}, \tag{20}$$

where, A_j is the j^{th} lowest-cost relevant item in the collection and $\{A_j\}_{j=1}^{\min(|A|,|L|)}$ is the set of the $\min(|A|, |L|)$ lowest-cost relevant items. The numerator is, therefore, the number of found lowest-cost items, and the denominator is the number of found items.

When evaluating to a depth of D, $P_c@D$, it is tempting, but wrong, to simply mark all items $\{A_d\}_{d=1}^{D}$ as relevant and all items $\{A_d\}_{d=D+1}^{\infty}$ as non-relevant. Doing so produces the wrong answer because $|L|$ could be smaller than D. We illustrate this by considering a results list of length $|L| = 2$ being evaluated to a depth of $D = 4$ (P_c@4). If the items in the results list are the 3rd and 4th cheapest items then the P_c score would be the same as that for the cheapest and 2nd cheapest items (which is intuitively incorrect). Indeed, the first stage of the ranking pipeline might be returning lists of different lengths, some less than D, for efficiency reasons as suggested by Culpepper, Clarke and Lin (2016). A worked example is provided in Table 4, which also shows that set-based (not score-based) precision is unable to distinguish between the case in the middle and case on the right.

Next we evaluate the metrics we have introduced by comparing them to previously introduced metrics.

7. Evaluation

In this section we evaluate buying power, buying power for K, selling power, and cheapest precision by scoring runs submitted to the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task".[3] We compute

[3] https://sigir-ecom.github.io/ecom2019/data-task.html

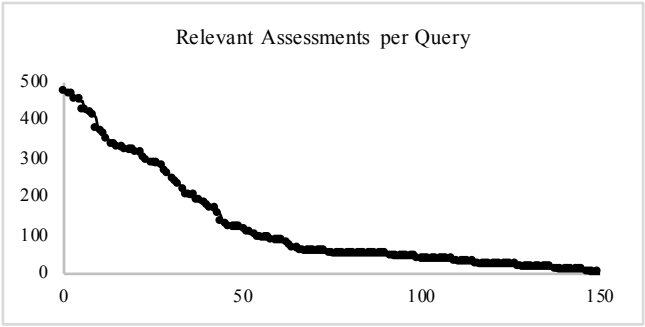


Figure 4: Number of *relevant* assessments per query for the 150 queries in the data set ordered from most to least. Largest is 472, smallest is 4, the median is 53.

the rank order of the runs using these metrics and a set of well known metrics (F_1, AP, RR, ESL, and $RBP_{p=0.95}$), and correlate these with each other using Spearman's Rank Correlation (ρ). We do not compare to others (RBP, adaptive cost models, etc.) because they are less commonly used.

7.1. Data

We use the data and runs from the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task". The task was described as "identifying the items to show when using non-relevance sorts". That is, given a document collection and a query, divide the document collection into two sets: Relevant, and not relevant. In short, build a binary classifier. Although this appears to be an unusual formulation of a ranking problem, eBay was investigating the first layer of their ranking pipeline. In their pipeline the first layer returns a list of candidate documents to be ranked further up the pipeline (either by best match, date, price, or otherwise).

For the challenge, eBay released a set of 899,681 documents along with 150 queries mined from their query logs. A total of 44,049 binary relevance assessments were provided to us by the challenge organisers. Of those, 18,128 were relevant and 25,921 were non-relevant. Figure 4 shows the number of relevant assessments for each query ordered from most to least. The number of relevant assessments ranges from 4 to 472, with a median of 53, a mean of 121, and a standard deviation of 130.

The exact details of the assessment process are not available (and presumed to be a corporate secret). What is known is that eBay has a model user described in their assessment guidelines and that the assessor makes decisions based on a set of rules described in the guidelines. In the case of ambiguity, the guidelines are updated to ensure consistent and explainable behavior and the assessor assesses the ambiguous document based on the new guideline (Goldberg et al., 2018). What is also known is that eBay values the traditional IR definition of relevance (fulfilling user need) over short term profits. Their reasoning is that a satisfied user will repeatedly return and spend more over a longer period of time than a dissatisfied user who might spend only once.

We obtained, from the challenge organisers, the anonymised best-run from each participating group (as evaluated in the final phase).[4] This gave us a total of 14 runs, each containing classifications for all 150 queries. 3 of the runs contained classifications for only 899,287 documents while the other 11 contained classifications for all 899,681 documents. We assume missing classifications occur when a document is not relevant to any query. For each query of each run we converted the set of relevant documents into an increasing-price ordered list by ordering on item price, which we extracted from the documents.

The challenge was divided into three phases, an unsupervised phase, a supervised learning phase, and a final evaluation phase. In preparing the runs we were careful to remove, from all results lists, all documents whose relevance was known during training. While this reduces the number of classifications in each run, it also prevents a run achieving a high score by simply placing all known-relevant documents in the run and nothing else.

Even though we do not know the fine details of this collection's construction, and in particular the assessment process, we do believe the collection is fit for purpose. It was constructed in order to study deterministic sort orders such as price-low-to-high, and it was constructed by an organisation that is tackling this problem (eBay). What is less

[4]There were more runs supplied to us than were submitted to the final phase because the organisers supplied the best run from each group regardless of whether or not it was submitted to the final phase

certain is whether or not the runs are suitable. We examine the runs in Section 7.3 where we present evidence that teams submitted runs favouring recall over precision.

We explored other data sets, including AmazonRel and GoogleLocalRec which were used by Nardini et al. (2019). Both of these sets contain only a single run and so are not suitable for a comparative analysis of the performance of multiple runs over multiple metrics. We are not aware of any other publicly available set of documents, assessments, and runs that could be used to evaluate metrics for eCommerce search.

7.2. Methods

The official challenge metrics were macro-precision, macro-recall, macro-F_1, micro-F_1, and *l2h_ndcg*. These were used to evaluate results sets, but we are interested in results lists and so we compare to macro metrics: F_1, *AP*, *RR*, *ESL*, and $RBP_{p=0.95}$.

To measure the performance at finding one relevant item, it is typical to use *RR* or *ESL*, and we now introduce *bp*. To measure the performance of finding K relevant items, RR_{k} was used, and we introduce *bp4k*. We set $K = 3$, assuming that the buyer is interested in purchasing 3 of a given item. This choice is arbitrary, but 10 queries have fewer than 10 relevant assessments (and 2 have the smallest, 4) so using more than 3 would result in some queries being unanswerable – and we do not wish to reduce the number of queries used in our experiments.

There is no obvious prior metric to *sp*. We set the number of slots, $|L|$, to 10, on the assumption that the seller is trying to fill a traditional SERP of 10 blue links.

Before P_{c}, the cost-agnostic precision metric, P, would have been used. We additionally compare our metrics to F_1, *AP*, and $RBP_{p=0.95}$ as they are commonly used. For these metrics we set the depth of the results list to the first quartile of the number of known-relevant items per query, i.e. 30. As this number increases, so to does the number of relevant documents in the cheapest set, eventually resulting in no difference between the cheapest n relevant documents and all relevant documents. Using 30 ensures that this happens in at most 25% of the queries (more precisely, 38 of the 150 queries).

We are interested in how well our metrics correlate with traditional cost-agnostic metrics. A perfect correlation would suggest that cost plays no part in the quality of a results list, which is counter intuitive when the results list is ordered on cost and might miss a relevant (and potentially low-cost) item. It is more reasonable to expect a strong, but not perfect, correlation. When ordered on cost, any relevant document will contribute to the metric's score, and any loss in score will be a function of the relative price difference between that item and any lower-cost item missed by the search engine. In other words, traditional relevance based metrics will give a very good indication of results list quality, but we posit that using cost will make it possible to make fine-grained distinctions that could not be made before.

The approach we take is to compute the rank-ordering of the runs for each metric and then compute the Spearman's Rank Correlation Coefficient (ρ) between the orderings. Spearman's ρ allows us to identify changes in the relative rank orders regardless of the actual score of the metric (which might not all be, for example, on a linear scale). We accept that there are other ways to correlate the performance of the runs, but we leave that for future work.

7.3. Results

Table 5 presents the Spearman Rank Correlation Coefficients for the runs using metrics based on the position of a single result in the results list: F_1, *AP*, *RBP*, *RR*, *ESL*, and our *bp*. Each is averaged over all 150 queries in the collection and to a results list depth of 30. We expect *bp* and *RR* to positively correlate, but to negatively correlate with *ESL*.[5] From the table, it is clear that there is a very strong correlation between all these metrics.

For a single query, *ESL* is the position of the first relevant item but *bp* is position and cost based. Imagine a single query in which two runs produce results lists with the same number, but different, relevant items at the same positions. If those items are cheaper in one run than the other, then the runs will score the same under *ESL* but not under *bp*. Averaged over a number of queries, one run might identify cheaper but fewer relevant items (scoring high in *bp*, but lower in *ESL*) while the other might identify more relevant but more expensive items (scoring lower in *bp*, but higher in *ESL*).

[5] Despite the *RR* for a single query being $\frac{1}{ESL+1}$, a perfect negative correlation is not seen because the scores are averaged over a set of queries, and one sequence is linear whereas the other is not. If a perfect negative correlation were always seen then it would not be possible to have two sequences such that $MRR_1 > MRR_2$ and $MESL_1 > MESL_2$ (when *RR* goes up, *ESL* goes down). As a counter example we present the locations of the first relevant result for two queries for system 1, {1, 4} and for system 2, {2, 2}. Here MRR_1=0.625, $MESL_1$=1.5, MRR_2=0.5, $MESL_2$=1. Fuhr (2018) observes that this is because $\mathbb{E}(\frac{1}{Rank}) \neq \frac{1}{\mathbb{E}(Rank)}$

Table 5
Spearman's ρ between rank orderings from metrics used to measure the quality of finding a single relevant item in the top 30 results.

	F_1	AP	$RBP_{p=0.95}$	RR	ESL	bp
F_1	1	0.9956	0.9956	0.9780	-0.9912	0.9692
AP	0.9956	1	1	0.9824	-0.9956	0.9780
$RBP_{p=0.95}$	0.9956	1	1	0.9824	-0.9956	0.9780
RR	0.9780	0.9824	0.9824	1	-0.9736	0.9956
ESL	-0.9912	-0.9956	-0.9956	-0.9736	1	-0.9692
bp	0.9692	0.9780	0.9780	0.9956	-0.9692	1

Table 6
Spearman's ρ between rank orderings from metrics used to measure the quality of finding 3 relevant items in the top 30 results.

	F_1	AP	$RBP_{p=0.95}$	$RR_{k=3}$	$bp4k_{K=3}$
F_1	1	0.9956	0.9956	0.9901	0.9901
AP	0.9956	1	1	0.9945	0.9945
$RBP_{p=0.95}$	0.9956	1	1	0.9945	0.9945
$RR_{k=3}$	0.9901	0.9945	0.9945	1	0.9868
$bp4k_{K=3}$	0.9901	0.9945	0.9945	0.9868	1

Table 7
Spearman's ρ between rank orderings from metrics used to measure the selling quality when 10 "slots" are available to be filled.

	F_1	AP	$RBP_{p=0.95}$	sp
F_1	1	0.9813	0.9813	0.9956
AP	0.9813	1	1	0.9769
$RBP_{p=0.95}$	0.9813	1	1	0.9769
sp	0.9956	0.9769	0.9769	1

AP gives higher scores the higher up the results list a relevant item is seen – with the amount diminishing quickly, but the score goes up as the number of relevant items in the results list increases. It strongly correlates with the single-item metrics as they all award higher scores for relevant items higher in the results list. *AP* correlates with F_1 because both increase as more relevant items (and fewer non-relevant items) are found in the top 30.

The correlation coefficients for purchasing multiple items (in this case 3) are presented in Table 6. A similar picture is seen to that of purchasing a single item: $bp4k_{K=3}$ very strongly correlates with $RR_{k=3}$. This is for the same reason, one run might be good at identifying relevant items, while another is better at choosing relevant and low-cost items. Correlation of F_1, *AP*, and *RBP* is discussed in the previous paragraph.

Table 7 presents the correlations between the metrics used to measure the quality of the site at selling. In this case the quality is measured at 10 results because that is the typical length of a SERP – and we assume a user will look at the results page and if they don't see what they are looking for they will reformulate their query.

To the best of our knowledge there are no preexisting metrics to measure seller quality and so we compare to F_1, *AP*, and *RBP*. The correlations are very strong, but the three metrics are measuring different things. F_1 is measuring the harmonic mean of the set-based precision and recall – on the assumption that the results are a set rather than a list. *AP* and *RBP* are measuring the ability of the search engine to place relevant items high in the list. Selling power, *sp*, is measuring how well the store can put the cheapest relevant items in the list – set and cost based. There is a clear relationship between these three metrics and so they very strongly correlate, despite measuring subtly different things.

Table 8 presents the correlations between metrics that might be used to measure the quality of early stages in the ranking pipeline. It unexpectedly shows that, in this data set, F_1 perfectly correlates with P – and so we investigated further. In Figure 5 we plot the precision at 30 documents against the recall at 30 documents. The figure shows an

Table 8
Spearman's ρ between rank orderings from metrics used to measure the quality of a result *set*, in this case, of 30.

	F_1	AP	$RBP_{p=0.95}$	P	P_c
F_1	1	0.9956	0.9956	1	1
AP	0.9956	1	1	0.9956	0.9956
$RBP_{p=0.95}$	0.9956	1	1	0.9956	0.9956
P	1	0.9956	0.9956	1	1
P_c	1	0.9956	0.9956	1	1

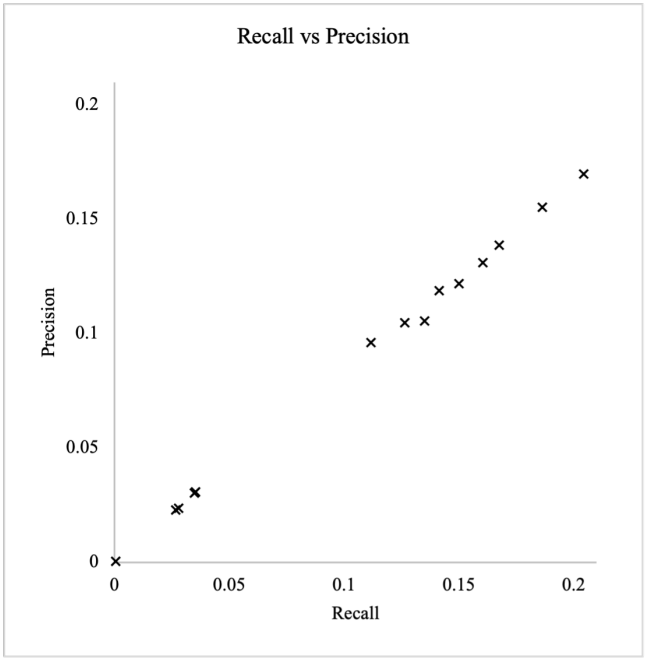


Figure 5: Run precision at 30 against run recall at 30 shows a linear relationship (Pearson = 0.999, Spearman = 1) between precision and recall at a fixed cut off of 30.

almost perfect straight line. Indeed, at a fixed and low cut-off, if the recall goes up then a non-relevant document must be replaced by a relevant document – and so we see a very strong correlation between precision and recall, and consequently F_1.

The correlation to P_c is perfect for a different reason. Figure 6 plots the precision of each run against the recall for that run (not cut off at 30). It can be seen that in all 14 runs the recall is high and the precision is low. This suggests that the approaches taken by the teams was to favor recall at the expense of precision, possibly because the original task was a classification task. Since in all cases the recall is high, the probability of having the lowest-cost relevant items in the results is high. Indeed, the correlation in Table 8 shows that the number of lowest-cost relevant items found is directly proportional to the number of relevant items found

We also correlated the runs across all the metrics we introduce. Table 9 presents the correlation coefficients. From this table it can be seen that, although the correlations are very strong, each metric is, indeed, measuring something slightly different. Indeed, P_c is a set based metric, *sp* is measuring the cost-effectiveness of each slot in a results list, *bp* is measuring the user's buying power at the first relevant item, and *bp4k* is measuring the user's buying power for more than one relevant item.

In this section we have shown that the new metrics we introduce correlate strongly with preexisting metrics, but not perfectly. They are measuring something different. Early information retrieval metrics such as precision, recall,

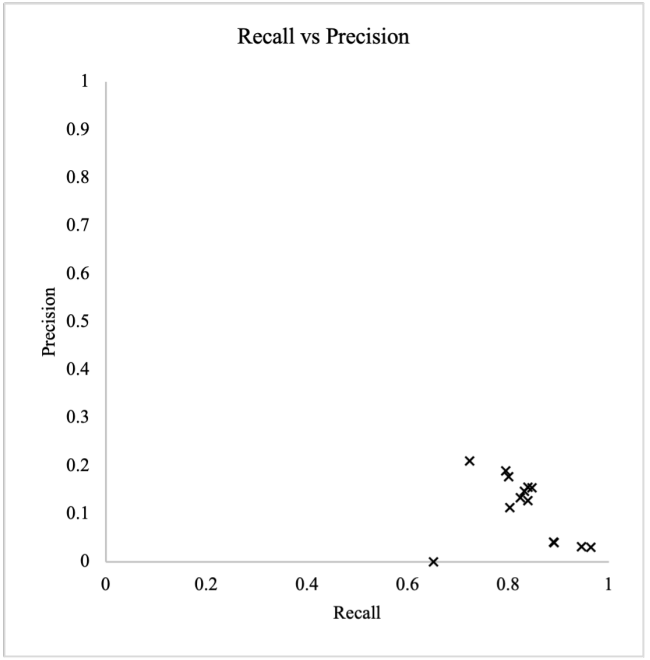


Figure 6: Run precision against run recall (with no cut off) shows a negative correlation (Pearson = -0.316, Spearman = -0.481) between recall and precision.

Table 9
Spearman's ρ between the metrics we introduce with a result list length of 30.

	P_c	sp	bp	$bp4k_{K=3}$
P_c	1	0.9956	0.9692	0.9901
sp	0.9956	1	0.9648	0.9945
bp	0.9692	0.9648	1	0.9725
$bp4k_{K=3}$	0.9901	0.9945	0.9725	1

and F_1 measure the quality of a *set* of results. Current metrics measure the quality of a *list* of results. The metrics we introduce measure the *cost of a list* of results.

8. Re-evaluation

In this section we re-examine the results of the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task". The official metrics of the challenge were: macro-precision, macro-recall, macro-F_1, micro-F_1, and *l2h_ndcg*. Almost all of those metrics measure search engine quality by averaging over the number of queries, but micro-F_1 does not, and so we do not generate scores using it.

As we do in Section 7, we measure the quality of the runs at a results list cut-off at 30. When looking for more than one item we are looking for 3. We have anonymised runs and so we refer to run numbers, not names. As we do not have the *final* run from each team, we reevaluate each run and do not rely on any previously published scores. Prior to doing so we removed, from all runs, those documents whose relevance was known at training time.

Table 10 lists the score each run would have achieved using the macro-precision, macro-recall, and macro-F_1, *l2h_ndcg* as well as the macro averaged scores for the metrics we introduce. The table shows that the best run is the best regardless of how it is measured, the worst run is the worst regardless of how it is measured, but in the middle there are runs that change order.

Table 10
Performance (at cutoff of 30 items) of the 14 runs ordered by F_1@30. The highest and lowest scoring runs are so, regardless of the metric. There is some re-ordering between positions 3 and 13.

	P		R		F_1		bp		$bp4k_{K=3}$		sp		P_c		$l2h\_ndcg$	
Team	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank
9	0.1698	1	0.2043	1	0.1793	1	0.2977	1	0.1803	1	0.1460	1	0.0813	1	0.1967	1
6	0.1553	2	0.1862	2	0.1638	2	0.2757	2	0.1644	2	0.1374	2	0.0724	2	0.1775	2
3	0.1387	3	0.1676	3	0.1468	3	0.2397	5	0.1407	4	0.1242	3	0.0676	3	0.1624	3
2	0.1311	4	0.1604	4	0.1392	4	0.2708	3	0.1411	3	0.1207	4	0.0642	4	0.156	4
5	0.1218	5	0.1500	5	0.1296	5	0.2410	4	0.1343	5	0.1116	5	0.0604	5	0.1443	5
4	0.1189	6	0.1414	6	0.1254	6	0.2119	8	0.1216	6	0.1054	6	0.0582	6	0.1356	6
1	0.1056	7	0.1349	7	0.1130	7	0.2216	6	0.1045	8	0.0963	8	0.0500	7	0.1266	7
8	0.1049	8	0.1265	8	0.1108	8	0.2176	7	0.1115	7	0.1002	7	0.0482	8	0.1241	8
7	0.0960	9	0.1117	9	0.1003	9	0.1814	9	0.0998	9	0.0853	9	0.0384	9	0.1073	9
12	0.0307	10	0.0352	10	0.0319	10	0.0594	10	0.0241	10	0.0276	10	0.0122	10	0.0338	10
13	0.0302	11	0.0347	11	0.0315	11	0.0589	11	0.0241	10	0.0271	11	0.0120	11	0.0334	11
10	0.0236	12	0.0278	12	0.0247	12	0.0474	13	0.0189	12	0.0208	12	0.0104	12	0.0264	12
11	0.0229	13	0.0267	13	0.0239	13	0.0484	12	0.0171	13	0.0200	13	0.0102	13	0.0259	13
14	0.0004	14	0.0004	14	0.0004	14	0.0017	14	0.0000	14	0.0004	14	0.0000	14	0.0005	14

Table 11
One-tailed paired t-test for top 5 runs under F_1@30. The p values are shown.

	Team 2	Team 3	Team 5	Team 6	Team 9
Team 2	-				
Team 3	0.2922	-			
Team 5	0.0642	0.0457	-		
Team 6	0.0193	0.0225	0.0001	-	
Team 9	0.0011	0.0004	0.0000	0.0007	-

Figure 7 presents the rank orderings or each run under each metric plotted against the rank order using F_1 (lower is better). Deviations from the diagonal are cases where a run has changed rank-order with F_1. For example, the third best run under F_1 is fifth best under bp.

According to Figure 7 and Table 10, team 9 produced the best run under all metrics and team 6 placed second. The third, fourth, and fifth positions depend on the metric used, with deviation from agreement only seen in the buying power metrics bp and $bp4k$. To explore this we conducted 1-tailed paired t-tests for the top 5 runs, testing for significance at the $\alpha = 0.05$ level. We report the p values before adjustment (under the Bonferroni correction, $\alpha = 0.0025$ as we run 20 tests).

For F_1@30, the p-values are shown in Table 11 from where it can be seen that team 9 submitted a run that is statistically significantly better then the others, team 6's run is statistically significantly better than the runs it beat, and teams 3, 2, and 5 submitted runs that are not statistically significantly better than each other (although 3 can only just be distinguished from 5). This suggests the results should be team 9 first, team 6 second, and teams 2, 3, 5 third equal.

For bp, the p-values are shown in Table 12. From that table, team 9 and team 6 submitted runs not statistically significantly different from the run from team 2. The runs from team 3 and team 5 are not statistically significantly different from each other. This suggests the result should be teams 2, 6, and 9 placed first with teams 3 and 5 placing second.

An example of how our metrics differ from prior metrics is exemplified in query 72 ("pig match") and the runs from teams 1 (Table 14) and 8 (Table 15). The assessments for this query are shown in Table 13. When searching for one item, $K = 1$, the runs are equivalent because both runs put the known cheapest item at the top of the results list. When looking for two items, $K = 2$, the run from team 1 is superior because it has the two cheapest items in positions 1 and 2, whereas the run from team 8 has 2 non-relevant items at positions 2 and 3, before the second cheapest item at position 4. However, if the user is trying to purchase 3 items then the rank order of the runs changes and team 8's run is superior because team 1 fails to identify the third cheapest item, which is identified by team 8. That is, team 1 shows

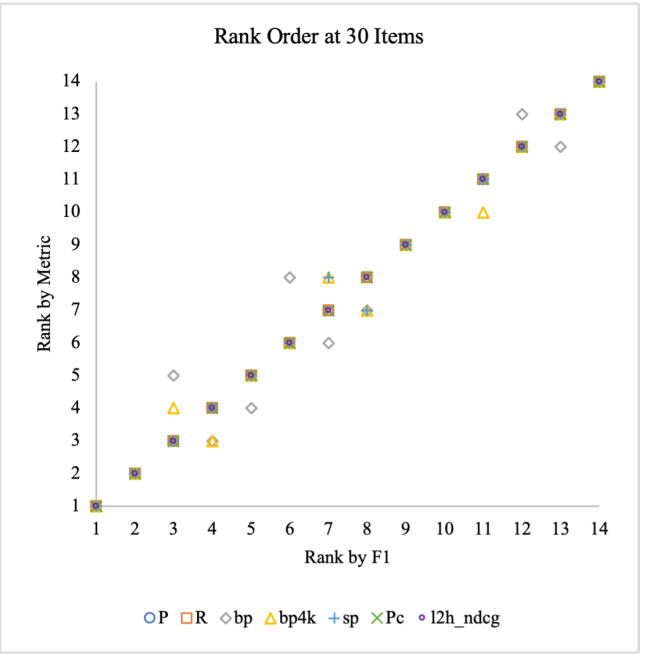


Figure 7: Rank order of the 14 runs using each of the metrics plotted against rank order using the official F_1 metric. Deviation from the diagonal are cases where the metric does not agree with F_1, and is therefore measuring something different: relevance with cost.

Table 12
One-tailed paired *t*-test for top 5 runs under *bp*. The *p* values are shown.

	Team 2	Team 3	Team 5	Team 6	Team 9
Team 2	-				
Team 3	0.0257	-			
Team 5	0.0485	0.4756	-		
Team 6	0.4055	0.0290	0.0100	-	
Team 9	0.0634	0.0008	0.0002	0.0008	-

a relevant item at $39.95 whereas team 8 shows a relevant item at $8.99! The *bp4k* scores for the two runs are shown in Table 16 for varying $K = 1$ to 6. When using Average Precision to measure the quality of the runs, the run from team 1 is never worse than the run from team 8. The AP scores for this query at the first 10 positions in the results list are shown in Table 17. In summary, *MAP* gives the impression that the run from team 1 is always better than the run from team 8, whereas if the user is trying to purchase 3 or more items, then team 8 has the better run.

9. Discussion

Up to this point we have assumed that cost is financial and that smaller is better. However, cost need not be financial. A user performing a point-of-interest (POI) search for a McDonald's may be more interested in distance than price. Such an interaction might be a search for the nearest McDonald's, only to discover that it is closed, followed by looking at the second result in the result list (the next nearest McDonald's), and so on. In this case distance from the current location, varying at each point in the results list, could be used as the cost – that is, the cost of an item might be dependant on previous items in the list.

With the buying power metric, the cost of the first relevant item is the sum of the costs up-to and including the first relevant item. For example, if the user searched for cornflakes and was presented with a result list showing tuna first and cornflakes second, then the user is assumed to have paid for both the tuna and the cornflakes. If the cost is distance, as it is in our McDonald's example, then this is equivalent to saying that the user must travel to McDonald's in order to

Table 13
The 11 lowest-cost known-relevant results for query 72.

Relevant	Doc ID	Price	Title
Y	1197502	4.50	Russia Match Box Label ! russland sssr animals pig cow corn apple N9
Y	1243420	5.99	Japanese Match Box Label Art Pig Boar Hog Snake 4 Postcard Lot Japan Asia Tokyo
Y	1735465	8.99	1940-50s PICTORIAL TOKEN...GUS' GOOD FOOD, CHICAGO - PIG HEAD-TAIL MATCHING COIN
Y	1149253	11.99	Antique Vintage Ceramic Lamb & Pig Matching Planters Farm Animals Lot of 2!
Y	1553440	19.14	2 Occupied Japan Pig Pushing Cart & Dog Match Holder Toothpick Whimsical Piggy
Y	1477023	30.69	Vintage Philippines hand Carved Wood ashtray & Match holder Native Tribal Pig
Y	1192466	39.95	Porcelain Pig Match Holder #4384
Y	1359398	39.99	Vintage Pig Match Holder at Heavy Ashtray Metal with Copper Finish 1940s
Y	1556658	64.95	Antique German Bisque Banker Pig Match Holder Swine # 1947 money bags 6" Tall
Y	1676367	65.00	Vintage Cast Iron Pig Match/Trinket Box
Y	1622992	75.00	BRASS VESTA MATCH SAFE CASE IN FORM OF A PIG

Table 14
The results of team 1 on query 72, truncated to 10 results.

Relevant	Doc ID	Price	Title
Y	1197502	4.50	Russia Match Box Label ! russland sssr animals pig cow corn apple N9
Y	1243420	5.99	Japanese Match Box Label Art Pig Boar Hog Snake 4 Postcard Lot Japan Asia Tokyo
N	1260792	12.99	Guinea Pigs Cricket Match RPPC Dressed Animals Sports Fantasy Real Photo 1957
N	1181152	24.95	Fancily DRESSED PIGS - Father BOAR Matches Pig SON to PIG BEAUTY ca1901 Postcard
N	1685732	31.13	Vintage Japan Porcelain PIG AND 2 PIGLETS Match Holder Ashtray
Y	1192466	39.95	Porcelain Pig Match Holder #4384
Y	1359398	39.99	Vintage Pig Match Holder at Heavy Ashtray Metal with Copper Finish 1940s
Y	1556658	64.95	Antique German Bisque Banker Pig Match Holder Swine # 1947 money bags 6" Tall
Y	1676367	65.00	Vintage Cast Iron Pig Match/Trinket Box
Y	1622992	75.00	BRASS VESTA MATCH SAFE CASE IN FORM OF A PIG

Table 15
The results of team 8 on query 72, truncated to 10 results.

Relevant	Doc ID	Price	Title
Y	1197502	4.50	Russia Match Box Label ! russland sssr animals pig cow corn apple N9
N	1533320	4.98	WHITE BLUFF, TENNESSEE: CARL'S PERFECT PIG BBQ RESTAURANT (PIG) -H
N	1181241	5.50	VINTAGE 3 PIG POSTCARDS by artist R L Wells, 2 matching undivided back, 1 divid
Y	1243420	5.99	Japanese Match Box Label Art Pig Boar Hog Snake 4 Postcard Lot Japan Asia Tokyo
N	1198813	6.17	Vintage Lot 3 Refrigerator Magnets Animals Pig Horse Tropical Fish Beach Ocean
N	1393851	7.99	Vintage Hand-Painted Salt & Pepper Shakers Farm Pigs w/ Matching Dish (Mexico)
Y	1735465	8.99	1940-50s PICTORIAL TOKEN...GUS' GOOD FOOD, CHICAGO - PIG HEAD-TAIL MATCHING COIN
N	1856398	9.40	Rare Vintage Girl Scout religious Sabbath matching pin and patch never used
N	1399810	9.99	Vintage Pottery Wall Hanging Match Holder with Pig On Front
N	1142263	10.00	Pig and Cow with matching Bandanas salt and pepper shakers.

discover that it is closed before then traveling to another McDonald's. In practice the total distance travelled to get to the first open McDonald's could be more than the sum of the distances in the initial results list, especially if the user is forced to backtrack past their starting point to get to the next nearest McDonald's. Assuming a user on a mobile device, re-searching at each closed store, the cost of the first item could be computed as the total distance travelled assuming the user is gaining knowledge at each point of interest (i.e. does a new search at each POI, but ignored results they've seen before).

Table 16
Buying power for K, *bp4k*, with different values of K calculated for the runs of team 1 and team 8 on query 72 showing that although team 1 is initially no worse, the better run depends on the value of K.

K	team 1	team 8
1	1	1
2	1	0.5002
3	0.1630	0.4415
4	0.1973	0.3253
5	0.2255	0.2846
6	0.2809	0.1449

Table 17
Top-10 Average Precision scores for team 1 and team 8 on Query 72 showing that when AP is used as the metric, team 1 is never worse than team 8.

Rank	Team 1	Team 8
1	1.0000	1.0000
2	1.0000	0.5000
3	0.6667	0.3333
4	0.5000	0.3750
5	0.4000	0.3000
6	0.4167	0.2500
7	0.4388	0.2755
8	0.4621	0.2411
9	0.4848	0.2143
10	0.5063	0.1929

Distance introduces a second difference from the way we introduce the metrics. Distance is likely to vary from user to user – whereas price should not (but price plus postage might). These metrics do not require the cost to be constant for all users, they only require the ability to compute it and to include it in evaluation.

Users of a job site may be interested in time as their cost function. They may be thinking "How quickly can I switch my academic job for a job in industry?". In this case time is of the essence, and could be used as the cost function.

Users of a traditional eCommerce search engine may be faced with the traditional price / time trade off. In this case cost can be expressed as a function of time and price. Some sites offer high-to-low price ordering. In this case the cost function might be the reciprocal of the price, or the highest possible item price minus the item price. Indeed, any cost function that is monotonic with user dissatisfaction could be used in our metrics. We leave it to future work to explore document features that might be used, and possible functions (linear or otherwise) of those features.

Our work is based on a number of assumptions stated in Section 2. We assume the assessments are sound and complete, and that the granularity is appropriate – but did not test this. Work on determining the suitability of assessments for a given task has a long history, is important, and is ongoing Piwowarski, Trotman and Lalmas (2008)Arabzadeh, Vtyurina, Yan and Clarke (2022).

We assume binary relevance and extending to graded relevance is left for future work. It seems obvious to simply take the graded assessment and multiply by the buying power (Equation 13 or Equation 15) and likewise for the other metrics. But this would require the additional assumption that there is a linear correlation between relevance and price (an item with a relevance of 0.5 at a price of 1 is exactly equal to an item with a relevance of 1 at a cost of 0.5). We have no reason to believe that this is the case. We believe this should be tested, and hence leave it for future work.

Our user is assumed to know what they are looking for – it is as if they have a shopping list that they must rigidly adhere to. Future work might account for a user who is browsing while shopping. In our running example, they might might be enticed by the tuna while shopping for the cornflakes. Or they might buy more cornflakes then they initially intended because of a "two for the price of one" deal the store offers. They might deliberately browse further down the results list than their item, just to see what else is on offer.

In Section 4.1 we suggested that prior metrics could be extended by using the value proportion defined in Equation 13. In our preliminary experiments we did exactly this. We started by defining $bpDCG$,

$$bpDCG = \sum_{i=1}^{n} \frac{\frac{A_1^c}{L_i^c}}{log_2(i+1)}, \tag{21}$$

where A_1^c is the lowest cost relevant item and L_i^c is the cost of item i in the results list of length n. The buying-power enhanced *nDCG*, *bpnDCG*, was calculated from *bpDCG* and *bpIDCG* (the ideal gain vector for the results list). When tested on the data set we are using, a strong correlation to the prior metrics was seen. As these experiments were preliminary, we did not experiment with different cut-off values for *bpnDCG*, and we did not experiment with searching for K of a particular item. That is, we used the entire results list rather than substituting n for Equation 16's m. Performing this substitution would adjust the user model from that of a user who always reads up to some fixed-cutoff (or the end of the results list, which ever is the sooner) to that of a user who is satisfied once they identify the K items they are trying to purchase. Further experiments with *bpnDCG*, and of cost enhanced versions of *MAP* and other metrics is left for future work.

Measuring the quality of any search engine is a complex task, and especially so in eCommerce where there are many stakeholders. These stakeholders might be buyers, sellers, advertisers, management, or others. In this contribution we have introduced a set of metrics that can be used to measure performance for buyers, sellers, and systems developers, in what we envisage as a Cranfield setting. Such metrics should be used in conjunction with other metrics (click-through rates, purchasing patterns, etc.) and A/B testing or interleaving to get a fuller picture of what the users are doing; and this information used along with management decisions to improve the quality of a site.

10. Conclusions

In this contribution we have suggested that cost as well as relevance should be used to measure the performance of search engine results lists that have been sorted deterministically. We introduced user-centric (*bp*, *bp4k*), seller-centric (*sp*), and system-centric (P_c) metrics that use cost and relevance to measure search results quality.

Using our new metrics we re-evaluated the best runs from each of the 14 groups who submitted to the "eBay SIGIR 2019 eCommerce Search Challenge: High Accuracy Recall Task". Our results show small changes in the rank orderings of these runs. From those runs we presented an example in which the inclusion of cost changed the relative rank order of the two runs – one run including cheaper items than the other.

Finally we discuss cost functions other than financial cost including: time, distance, and a combination of price and time.

We believe that with metrics such as these we introduce, it will be possible to accelerate research into deterministic sort ordering of results lists – a problem that has previously been identified and tackled by others but requires more work to bring to the quality of best-match ordering.

CRediT authorship contribution statement

Andrew Trotman: Conceptualization of this study, Methodology, Software, Writing - Original draft preparation. **Vaughan Kitchen:** Software, Writing - Original draft preparation.

References

Arabzadeh, N., Vtyurina, A., Yan, X., Clarke, C.L.A., 2022. Shallow pooling for sparse labels. IRJ .

Azzopardi, L., 2014. Modelling interaction with economic models of search, in: SIGIR 2014, p. 3–12.

Azzopardi, L., Thomas, P., Craswell, N., 2018. Measuring the utility of search engine result pages: An information foraging based measure, in: SIGIR 2018.

Bailey, P., Moffat, A., Scholer, F., Thomas, P., 2015. User variability and IR system evaluation, in: SIGIR 2015, p. 625–634.

Bhargav, S., Sidiropoulos, G., Kanoulas, E., 2022. 'It's on the tip of my tongue': A new dataset for known-item retrieval, in: WSDM 2022, p. 48–56.

Cooper, W.S., 1968. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. American Documentation 19, 30–41.

Culpepper, J.S., Clarke, C.L.A., Lin, J., 2016. Dynamic cutoff prediction in multi-stage retrieval systems, in: ADCS 2016, pp. 17–24.

Degenhardt, J., Kallumadi, S., Porwal, U., Trotman, A., 2019. ECOM'19: The SIGIR 2019 workshop on eCommerce, in: SIGIR'19, pp. 1421–1422.

Fuhr, N., 2018. Some common mistakes in IR evaluation, and how they can be avoided. SIGIR Forum 51, 32–41.

Goldberg, D., Trotman, A., Wang, X., Min, W., Wan, Z., 2018. Further insights on drawing sound conclusions from noisy judgments. TOIS 36, 36:1–36:31.

Iyengar, S.S., Lepper, M.R., 2000. When choice is demotivating: Can one desire too much of a good thing? Journal of Personality and Social Psychology 79, 995–1006.

Järvelin, K., Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. TOIS 20, 422–446.

Jiang, J., Allan, J., 2017. Adaptive persistence for search effectiveness measures, in: CIKM 2017, p. 747–756.

Kelly, D., Azzopardi, L., 2015. How many results per page?: A study of SERP size, search behavior and user experience, in: SIGIR 2015, pp. 183–192.

Moffat, A., Zobel, J., 2008. Rank-biased precision for measurement of retrieval effectiveness. TOIS 27, 2:1–2:27.

Nardini, F.M., Trani, R., Venturini, R., 2019. Fast approximate filtering of search results sorted by attribute, in: SIGIR 2019, pp. 815–824.

Piwowarski, B., Trotman, A., Lalmas, M., 2008. Sound and complete relevance assessment for xml retrieval. TOIS 27.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M., 1994. Okapi at TREC-3, in: TREC-3, pp. 109–126.

Smucker, M.D., Clarke, C.L., 2012. Time-based calibration of effectiveness measures, in: SIGIR 2012, p. 95–104.

Spirin, N.V., Kuznetsov, M., Kiseleva, J., Spirin, Y.V., Izhutov, P.A., 2015. Relevance-aware filtering of tuples sorted by an attribute value via direct optimization of search quality metrics, in: SIGIR 2015, pp. 979–982.

Trotman, A., Degenhardt, J., Kallumadi, S., 2017. The architecture of eBay search, in: The SIGIR 2017 Workshop on eCommerce.

Trotman, A., Kallumadi, S., Degenhardt, J., 2018. High accuracy recall task, in: The SIGIR 2018 Workshop on eCommerce.

Zhang, F., Liu, Y., Li, X., Zhang, M., Xu, Y., Ma, S., 2017. Evaluating web search with a bejeweled player model, in: SIGIR 2017, p. 425–434.