Diversifying Web Search Results

Davood Rafiei*
University of Alberta
drafiei@ualberta.ca

Krishna Bharat
Google Inc.
krishna@google.com

Anand Shukla
Google Inc.
anand@google.com

ABSTRACT

Result diversity is a topic of great importance as more facets of queries are discovered and users expect to find their desired facets in the first page of the results. However, the underlying questions of how 'diversity' interplays with 'quality' and when preference should be given to one or both are not well-understood. In this work, we model the problem as expectation maximization and study the challenges of estimating the model parameters and reaching an equilibrium. One model parameter, for example, is correlations between pages which we estimate using textual contents of pages and click data (when available). We conduct experiments on diversifying randomly selected queries from a query log and the queries chosen from the disambiguation topics of Wikipedia. Our algorithm improves upon Google in terms of the diversity of random queries, retrieving 14% to 38% more aspects of queries in top 5, while maintaining a precision very close to Google. On a more selective set of queries that are expected to benefit from diversification, our algorithm improves upon Google in terms of precision and diversity of the results, and significantly outperforms another baseline system for result diversification.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Algorithms, Theory, Experimentation, Measurement

Keywords

Search diversity, Result diversity, Query ambiguity

1. INTRODUCTION

With everyday increases to the size of the World Wide Web and the potential matches of queries, selecting "best" matches for the top few slots of a result page is becoming more constrained. At the same time, users expect to find their relevant matches in the first page (if not in top 5 or 3), but as the diversity of users increases, so does their needs whereas the number of slots in the first page remains fixed. To have a better understanding of the challenges, consider the following scenarios.

EXAMPLE 1. *Consider a scenario where the result set* $\{A, B, C, D, E, F, G, H, I, J\}$ *is retrieved, and the query is likely to be posed by one of four hypothetical users. Suppose one such user (if posed the query) would find* `A` *and* `D` *relevant to his/her search; the second user would find* `G` *and* `H` *and the third user would find* `A` *and* `E` *relevant to their searches. The fourth user would find no relevant document in the returned set. An interesting question is as both the number of likely users and the size of the result set increases, what is the best strategy for ordering the results such that many users would find their relevant pages in the top few slots.*

The problem is often more complex, and the search experiences of users generally vary depending on where relevant documents are shown in result pages. In particular, results are often browsed from top to bottom, and it requires less effort to find a relevant document, for instance, in position one, compared to other positions. Also given an ordering, the more users find their relevant documents in top positions, the better score should be given to the ordering. In other words, the overall search performance depends on both the number of users and the quality of the search experience of each user.

EXAMPLE 2. *Consider the result set in Example 1 but suppose this time a reward is paid when relevant documents are retrieved. Let the amount of payoff be proportional to the reciprocal ranks of the relevant documents, i.e. the payoff at rank* i *is* $1/i$ *if the document is relevant and* 0 *otherwise. Suppose the payoff for a result ordering is the mean payoff for all relevant documents. For instance, if only the first and the fourth documents are relevant to a particular search, the mean payoff would be* $(\$1.00 + \$0.25)/2 = \$0.62$. *Now consider a search scenario where in 45% of the cases* $\{A, D\}$ *is relevant, in 12% of the cases* $\{G,H\}$ *is relevant, in 1% of the cases* $\{A, E\}$ *is relevant and in the remaining 42% of the cases, none of the documents are relevant. Table 1 gives both the payoff and the weighted payoff when the results are ordered from A to J. Under the given workload, the mean payoff is estimated to be \$0.31 for the given ordering.*

In real settings, typically very little is known about a large class of queries in advance. According to some estimates, about 60% of search engines queries are new [16]. In cases where the queries are known, it is also generally difficult to

*The work is done while the author was visiting Google in Mountain View, CA.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

		45%	12%	1%	42%	Sum 100%
$1.00	A	A		A		
$0.50	B					
$0.33	C					
$0.25	D	D				
$0.20	E			E		
$0.17	F					
$0.14	G		G			
$0.12	H		H			
$0.11	I					
$0.10	J					
payoff		$0.62	$0.13	$0.60	$0.00	
weighted		$0.28	$0.02	$0.01	$0.00	$0.31

Table 1: Example scenario

identify in advance which subsets of the results would be relevant to which classes of searches and the frequency of searches in each class. Hence workloads and query classes are rarely known in advance. In this paper, we base our analysis on click-through rate estimates and statistics about search results, mainly because click-through rates, unlike user workload and query statistics, can be estimated with a good accuracy (e.g. [15]). If clicks are treated as votes for relevance, the mean and the standard deviation of this relevance can be estimated; this leads to further statistics about a result set or any subset of the result set.

As our contribution, we formalize the problem of diversifying search results and propose solutions to address it. More specifically, we cast the problem as an optimization task and present algorithms to estimate the optimization parameters. We report on the feasibility and the accuracy of our estimations using various data sources. We further conduct experiments using queries from different sources, including Wikipedia pages [19] and search engine logs, and report the effectiveness of our algorithm in diversifying Google search results. Our approach fits within the general risk minimization framework of Zhai and Lafferty [21] in that a risk function is minimized and a variable capturing user behaviour is introduced. Our work focuses on one particular loss function which is the variance of relevances.

The rest of the paper is organized as follows. In the next section, we formalize the problem and our approach to diversifying search results. The presented model has a few input parameters; Section 3 presents a few algorithms for estimating these parameters. Section 4 presents our experimental evaluation of the work. Related works are discussed in Section 5 and Section 6 concludes the paper.

2. PORTFOLIO MODEL OF SEARCH

Let S be a result set and Q_S be a set of *searches* for which S is retrieved. Our notion of a *search* here includes all query aspects including spatial, temporal and lexical features. Hence all searches are considered unique, meaning the same query expression issued by different users or even the same user at different times are considered different mainly because the relevant result sets can be different (e.g. the desired result for "Michael Jordan" can be in one search the NBA player and in another search the U. Berkeley Prof.). The problem of result diversification informally can be described as finding a set $S' \subseteq S$ such that S' includes relevant documents for as many queries as possible in Q_S. Naturally result diversification is meaningful when S is too large to be fully browsed. We would refer to set S' as a *portfolio*. In real settings, we may have some constraints on relatedness, diversity or the size of S' based on page layout and statistics on the number of pages browsed or clicked per query.

Let z_u be a random variable indicating the relevance of document u to queries in Q_S. Suppose z_u takes values in the range $[0, 1]$. Denote the expectation and the variance of z_u respectively by $E(z_u)$ and $\sigma^2(z_u)$. Now let $Z = [z_1, z_2, \ldots, z_n]^T$ be a vector of random variables indicating the relevance of the documents in S, as just described, where $n = |S|$. Denote the correlation between two variables z_i and z_j by ρ_{ij} and form the covariance matrix of the variables associated to the result set S. The covariance matrix is symmetric with entry at row i and column j set to $\sigma(z_i)\sigma(z_j)\rho_{ij}$.

More formally, let's indicate the inclusion of pages in a portfolio with a weight vector $W = [w_1, \ldots, w_n]^T$ where $0 \leq w_i \leq 1$ for $i = 1, \ldots, n$ and $\sum_i w_i = 1$. Given a weight vector W, the expected relevance of the portfolio to queries is $W^T E(Z)$ and the variance is $W^T C W$ where C is the covariance matrix of the result set [1]. It should be noted that the expectation here gives the precision of the results, averaged over queries in Q_S, and the variance indicates the degree of dispersion or variation in precision between queries.

Definition 1. *A portfolio is diversified if its expectation is relatively high and its variance is relatively low.*

This is a natural and intuitive definition of result diversity. Given the uncertainty around queries and the intentions of users posing them, we not only want to increase the average relevance or expectation of a portfolio but also want to reduce the variations on relevance between potential searches for which the portfolio may be returned. With this definition, consider two portfolios W_1 and W_2 and suppose W_1 has the same expectation as W_2 but a smaller variance. There are two possible cases where W_1 can have a smaller variance. First, variables in Z with large (small) variances may have overall smaller (larger) weights in W_1 compared to W_2; this combined with the fact that W_1 and W_2 have the same expectations would indicate that W_1 overall prefers documents that are relevant to a larger number of queries over those that are relevant to only a few; hence W_1 is more diverse. Second, variables in Z can have correlations and W_1 may give less weight to correlated variables than W_2 which can result in a smaller variance for W_1. This would in turn indicate that W_1 prefers less correlated and more diverse results. On the other hand, diversity is meaningful if the results are relevant or the expectation is not low.

2.1 Search Optimization

A search for a diversified result set can be modeled as an optimization problem where a portfolio is sought such that the portfolio variance is minimized while the expected relevance is fixed at a certain level e, i.e.

$$\min_W \frac{1}{2} W^T C W \tag{1}$$

subject to

$$W^T E(Z) = e, W^T 1 = 1 \tag{2}$$

[1]Throughout this paper, we assume W is a column vector and its transpose W^T is a row vector.

where W consists of all non-negative weights [2]. Suppose $E(Z)$ and 1 are linearly independent; if not, one of the constraints must be redundant and can be eliminated before solving the equation. For vector W^* to be a solution of Eq. 1 and 2, the necessary first-order condition is

$$CW^* - \lambda^* E(Z) - \gamma^* 1 = 0 \quad (3)$$

$$E(Z)^T W^* = e, 1^T W^* = 1 \quad (4)$$

where λ^* and γ^* are scalers (referred to as Lagrange multipliers).

2.1.1 Uniqueness of solution

To show the uniqueness of solution for our optimization, we first eliminate the constraints. If we denote $[E(Z)\ 1]^T$ and $[e\ 1]^T$ respectively by A and b, then the equality constraints in Eq. 2 can be written as $AW = b$. A general approach for reducing this constraint is to choose $Y \in \mathbb{R}^{n\times 2}$ and $V \in \mathbb{R}^{n\times(n-2)}$ such that $AV = 0$ and AY is invertible [3], and rewrite W as

$$W = Y(AY)^{-1}b + Vx_v. \quad (5)$$

Further details on finding Y and V can be found elsewhere (e.g. see [8]). It is easy to see that with the setting in Eq. 5, the constraint $AW = b$ holds for all values of x_v, and that the optimization problem in Eq. 1 can be equivalently expressed as an unconstrained problem after replacing W with its equivalent expression in Eq. 5.

DEFINITION 2. *A square matrix $M \in \mathbb{R}^{n\times n}$ is positive definite if $X^T MX > 0$ for all $X \in \mathbb{R}^n$, and positive semi-definite if $X^T MX \geq 0$.*

Equivalently, M is positive definite if and only if all of the eigenvalues are positive, and M is invertible if and only if all of the eigenvalues are non-zero (see, for example, [3]). Clearly a positive definite matrix is always invertible.

THEOREM 1. *If $V^T CV$ is positive definite, then there is a unique $(W^*, \lambda^*, \gamma^*)$ satisfying the conditions of Eq. 4 and W^* is the unique global solution of Eq. 1 and 2.*

PROOF. This is the direct consequence of Lemma 16.1 and Theorem 16.2 of Nocedal and Wright [12]. □

LEMMA 1. *Let C be a covariance matrix and $AV = 0$; $V^T CV$ is positive definite if no component of W is a linear function of other components.*

PROOF. The covariance matrix C is always positive semi-definite, i.e. $X^T CX \geq 0$ for all non-zero vectors X; otherwise the variance would be negative. Suppose there is a non-zero vector p such that $p^T Cp = 0$. Consider the random variable $pW^T = [p_1 W_1, \ldots, p_n W_n]$;

$$\sigma^2(pW^T) = (p^T Cp) = 0.$$

This means $pW^T = b$ for some constant b. Since p is non-zero, at least one component of p say p_1 must be non-zero. That means we can write W_1 as a linear function of other components of W, and this contradicts our assumption that no component of W is a linear function of other components. □

With the replacement of variables in Eq. 5, the preconditions of Lemma 1 and Theorem 1 hold, hence the uniqueness of an optimal solution is guaranteed. Our experiments with hundreds of thousands of queries, as discussed in Section 4, also confirmed that a unique solution is always reachable. Next section presents our methods for estimating the model parameters including $E(Z)$ and C.

3. ESTIMATING THE MODEL PARAMETERS

The optimization model, as presented in Section 2.1, has a few input parameters. Given a query and a set of matching documents, the relevance expectation for all matching documents, i.e. $E(Z)$, would have to be estimated before evaluating the expectation in Eq. 2. Furthermore, to construct the covariance matrix (in turn used in Eq. 1), the variance of relevances on each result document, and the pairwise correlations between relevances of result documents have to be evaluated. The data in our disposal for estimation is (a) click data which may be treated as votes for relevance, and (b) document content.

3.1 Correlations between pages

Consider the set of matching pages of a query, and let p and q be two arbitrary pages in the result set. We call p and q *positively correlated* if there is evidence that whenever p is relevant to a search, then q is also likely to be relevant and vice versa. Correlation statistics is important in optimizing search results, as evidenced in our formulation discussed in the previous section; but estimating correlation can be challenging mainly because *relatedness* and *relevance* are often subjective and may depend on queries. For example, `daimler.com` and `toyota.com` are related with respect to query "car makers" but not so related with respect to queries "german car makers" and "japan manufacturers".

One approach for estimating correlation is based on past search data; if two pages are frequently retrieved or clicked for the same query, it is likely that they would be retrieved or co-clicked in future. A problem though is that this data is very sparse. A large fraction of pages are never clicked or retrieved. Even at the site level, the data is still too sparse. In our experiment with 21 million lines of the AOL query log [13], the number of sites that were co-clicked was less than 9 million. After removing the pairs with frequency two or less and also those with confidence [4] less than 0.10, the number was dropped to 9700 pairs. This level of confidence was relatively low, but even at this level and with 1.2 million unique sites in the log, the chance of finding an estimate for an arbitrary pair was less than 0.00000001.

Another approach for estimating correlation is to use the textual contents of pages; if two documents are similar in their textual contents and one is relevant to a query, the other is also likely to be relevant. On the same basis, a large body of work in the IR community has studied different metrics for finding similar documents (e.g. [2]). As an

[2]Alternatively, one may fix the variance and maximize the expectation; because of the uniqueness of a solution (as shown next), for a fixed variance, the problem has a unique expectation and vice versa.

[3]A matrix $A \in \mathbb{R}^{m\times n}$ is *invertible* if there exists $A^{-1} \in \mathbb{R}^{n\times m}$ such that $AA^{-1} = I$ where I is the $m \times m$ identity matrix.

[4]For a pair s_1 and s_2 of frequencies respectively f(s_1) and f(s_2) and the joint frequency f(s_1,s_2), confidence(s_1,s_2)=f(s_1,s_2)/max(f(s_1),f(s_2)).

argument against applying this idea to Web pages, consider an obscure Web page that imitates a very well-known page and scores the highest degree of textual similarity. Hence textual similarity alone is not sufficient to establish relatedness. However, this is less of an issue in our case since correlation is only sought between pages that are retrieved (for a query) and generally obscure and low-quality pages have less chance of making it to a result set. Even if a low quality page makes to the result set, with a strong correlation between the two (high quality and a low quality) pages, the low quality page has less chance of making to the portfolio.

To further reduce noise and to keep the number of false correlations within some bound, we use a more salient set of features to indicate if pages refer to the same set of concepts and entities. Our set of features, in particular, included *entities*, *numbers*, *query extensions* and *site names*. Entities are identified using a simple heuristic that looks for capitalized terms and phrases in sentences. Query extensions are terms and phrases that appear in result pages and extend query terms. For example, 'city of palo alto', 'palo alto chamber of commerce' and 'palo alto restaurants' are all extensions of the query 'palo alto'. Query extensions are important for queries with multiple aspects, and may correlate pages on each aspect. Numbers are included since they give quantities such as phone number, zip code, year, height and weight; these quantities may relate pages that discuss the same entities or concepts. Extracted entities, query extensions and numbers were weighted based on their Inverse Document Frequencies [5] (IDF). Finally site names are also included based on the observation that pages on the same site are generally related. In fact, some search engines limit the number of pages from the same site (so-called site collapsing) as an attempt toward diversifying the results. This observation does not hold for large portals that host millions of pages such as `yahoo.com`. We deal with this problem by assigning a weight to each site which is inversely proportional to the number of pages on the site (similar to IDF weighting).

Consider two Web pages p and q in a document collection and denote their set of common features with F. If $P(f)$ denote the probability of observing feature f in an arbitrarily chosen document in the collection, and assuming independence of the features in F, the probability that feature set F (say of p) is found in an arbitrarily chosen document (including q) by chance is $\Pi_{f\in F}P(f)$. The larger this probability is, the smaller the correlation between p and q should be. Of course, correlation is bounded from upward to 1, and when p and q are independent, the correlation should be zero. Based on these observations, the correlation between p and q can be expressed as:

$$C(p,q) = \begin{cases} (1/\epsilon)(-log(\Pi_{f\in F}P(f))) & -log(\Pi_{f\in F}P(f)) < \epsilon \\ 1 & else \end{cases}$$

where ϵ is a threshold, in our case, set to $-log(1/N)$ and N is the number of documents in the collection; this setting indicates the point where the expectation of $\Pi_{f\in F}P(f)$ drops to one. The log function can be pushed in giving an equivalent expression for correlation as the (normalized) sum of the idf values for features in F.

$$C(p,q) = \begin{cases} (1/\epsilon)(\sum_{f\in F} -log(P(f))) & \sum_{f\in F} -log(P(f)) < \epsilon \\ 1 & else \end{cases} \quad (6)$$

[5] $idf(t) = -logf(t)/N$ where $f(t)$ is the number of documents that has t and N is the number of documents in the collection.

j	E($\hat{N}_j a$)	Var($\hat{N}_j$)
1	0.3075	0.2073
2	0.0792	0.0642
3	0.0522	0.0416
4	0.0343	0.0265
5	0.0258	0.0194
6	0.0198	0.0145
7	0.0159	0.0114
8	0.0135	0.0095
9	0.0124	0.0086
10	0.0129	0.0089
11	0.0023	0.0014
12	0.0018	0.0011

Table 2: Position payoff

3.2 Relevance expectation and variance

Consider a result set S and let u be a document in S. If Z_u denote the relevance (or payoff) of u to queries in Q_S, we want to estimate the expectation and the variance of Z_u. Assuming that the distribution of Z_u doesn't change much over time (e.g. Michael Jordan's NBA page remains relevant to more queries than the Berkeley Prof's home page), its expectation and variance can be estimated based on past queries. In particular, our estimation is based on the observation that the relevance of a document to a query is directly related to the number of clicks the document is expected to receive. Generally not all clicks are equally important; especially more relevant pages are likely to be clicked first. Also long clicks, where more time is spent on a page, may be considered more important than short ones. Hence query clicks may be ordered, and each click may be assigned a payoff proportional to its rank. Without loss of generality, suppose the amount of payoff is set to the reciprocal rank of a click. If the random variable Z_{uj} denote the payoff for document u at position j, then

$$Z_{uj} = \begin{cases} 1/i & url\ u\ at\ position\ j\ receives\ the\ ith\ click \\ 0 & else. \end{cases}$$

Similarly, the payoff at position j, denoted by N_j, can be written as

$$N_j = \begin{cases} 1/i & position\ j\ receives\ the\ ith\ click \\ 0 & else. \end{cases}$$

Table 2 gives the expectation and variance of this payoff for top 12 positions, as estimated from the AOL search log data [13]. On the other hand, Z_{uj} directly depends on both N_j, and the bias introduced by presenting u at position j. If random variable X_u denote this bias toward u, Z_{uj} can be expressed as

$$Z_{uj} = N_j + X_u. \quad (7)$$

Here $X_u = Z_{uj} - N_j$ gives the difference in payoff between the case where u is present at position j and the case where u is not. X_u can be estimated for each u based on estimates of Z_{uj} and N_j. It should be noted that in real settings Z_{uj} may also correlate (either positively or negatively) with

$Z_{u'j'}$ for pages u' shown at positions $j' < j$; this correlation is not easy to estimate and is not taken into account in our formulation. The expectation and the variance of Z_{uj} can be expressed as

$$E(Z_{uj}) = E(N_j) + E(X_u),$$
$$\sigma^2(Z_{uj}) = \sigma^2(N_j) + \sigma^2(X_u) + 2Cov(N_j, X_u) \quad (8)$$

where the last term in the expression of variance gives the covariance of N_j and X_u and is zero when N_j and X_u are independent. The expectation and the variance of N_j and X_u can be estimated in advance of queries and can be plugged in Eq. 8 at query time to derive estimates of the expectation and variance of Z_{uj}. As shown in Figure 1-a for sites in the AOL log data [6], the bias for 80% of the sites is either zero or -0.05 and for the remaining 20% of the sites it is distributed in the range from 0.05 to 0.7.

Our formulation of bias in Eq. 7 assumes that X_u is independent of the position, hence it gives the average bias over all positions where u appears. A breakdown of bias over positions, as shown in Figure 1-b for the AOL log data, reveals that X_u is not uniformly distributed over all positions, and rather it has a direct relationship to the positions where u appears. With the sparsity of data however, it is difficult to estimate the bias for each site and each position; there is little click data on many URLs, and the URLs that are clicked hardly appear in more than very few positions. To address the problem of sparsity, we use the distribution of bias over positions, estimated over all sites, to obtain an estimate of bias for a specific site on a specific position. In particular, given a site u and position i, assuming that the bias toward u at position i does not differ much from the bias distribution of other sites at the same location, X_u can be scaled according to the distribution, giving a more accurate estimate at position i. As shown in Figure 1-b for $E(X_u)$, the scaling ratio gets close to zero for $i > 10$.

3.3 Target expectation

One last parameter to our search optimization is a target level of expectation as denoted by e in Eq. 2. Assuming that the expected values of relevance or clicks per page are in the range $[0, 1]$, the target expectation is also constrained to the range $[0, 1]$. If we denote the greatest expectation of a result set S by $e_{max}(S)$, the target expectation for S also cannot exceed $e_{max}(S)$. For the boundary values, Eq. 1 has trivial solutions; more specifically when $e = e_{max}(S)$, an optimal portfolio includes only the document(s) with the greatest expectation(s), and for $e = 0$, an optimal portfolio includes no document.

Definition 3. *For a given result set S, relevance estimates are mean-variance efficient if for any pair Z_{uj} and Z_{vj} where $u, v \in S$, $E(Z_{uj}) \geq E(Z_{vj})$ iff $\sigma^2(Z_{uj}) \geq \sigma^2(Z_{vj})$.*

Mean-variance efficiency is expected to hold under natural settings of search engines; an efficient engine is expected to push URLs with high expectations up in the ranking until an equilibrium is reached between the expectations and the variances.

Theorem 2. *Assuming mean-variance efficiency of the individual estimates Z_{uj}, variance of an optimal portfolio is a monotonically increasing function of e.*

[6]Note that this dataset only has information about clicks; it is assumed that all sites have the same chance of being shown.

e		portfolio size	
relative	absolute (mean)	mean	var
MT(2)	0.38	2.39	1.83
MT(3)	0.29	3.01	2.02
MT(5)	0.20	3.61	2.11
MT(10)	0.12	3.20	2.02

Table 3: Portfolio size varying the target expectation

Proof. Suppose the result set includes a document d_0 with both expectation and variance zero; this can be, for example a document which is never shown hence it cannot receive a click. Denote the random variable indicating the relevance of d_0 by Z_0. Consider target expectations e_1 and e_2 where $e_1 > e_2$ and let W_1 and W_2 be the respective optimal portfolios. As a contradiction, suppose $\sigma^2(W_1) \leq \sigma^2(W_2)$. Let $\epsilon = (e_1 - e_2)/n$ for some positive integer n. Construct portfolio $W_1{}'$ as follows: first set $W_1{}' = W_1$, then iteratively select a document d_i with expectation greater than ϵ; denote the random variable indicating the relevance of d_i by Z_i. Reduce the associated weight w_i in $W_1{}'$ by $\epsilon/E(Z_i)$ and add $\epsilon/E(Z_i)$ to the weight of d_0 (to keep the norm of the weight vector 1). Repeat the iterative step until $E(W_1{}') = e_2$. Since initially $E(W_1{}') = e_1$ and $e_1 > e_2$, after n iterations (for an appropriate n) the termination condition would become true. With the new $W_1{}'$ and because of the mean-variance efficiency of estimates for individual documents, $\sigma^2(W_1{}') < \sigma^2(W_1)$. We also have $\sigma^2(W_1) \leq \sigma^2(W_2)$, hence $\sigma^2(W_1{}') < \sigma^2(W_2)$. But this is a contradiction since W_2 is an optimal portfolio for the target expectation e_2; this completes the proof. □

With a monotonic relationship, setting a target expectation is not straightforward since a higher expectation would also mean a higher variance. To find a trade-off, we further studied the portfolio size as the target expectation varied. Given statistics on the browsing behaviour of users and the fact that the vast majority of users only browse a few top-ranked results, it is reasonable to keep the portfolio size small. Although the size of a portfolio cannot be set directly, our experiments show that the portfolio size to some degree is a by-product of the target expectation. Table 3 shows mean portfolio size for 10,000 queries randomly selected from Google query log as the target expectation varies from MT(2) to MT(10) where MT(i) denotes the mean of top i expectations in a result set. The optimal portfolio size increases as the target expectation e decreases until it reaches $MT(5)$ after which there is a reduction in size. This is not surprising given that when $e = 0$, the optimal portfolio size is 0 (as discussed earlier). Examining the changes in optimal portfolios as e increases reveals that for smaller values of e the changes are in the form of including additional documents in the portfolio, whereas for larger values of e the changes are in the form of both additions and substitutions. Table 4 shows these changes for the same set of random queries. The changes to optimal portfolios are overall small, indicating the robustness of the portfolio selection to changes in e.

4. EXPERIMENTS

This section provides a preliminary evaluation of our search optimization and diversified results.

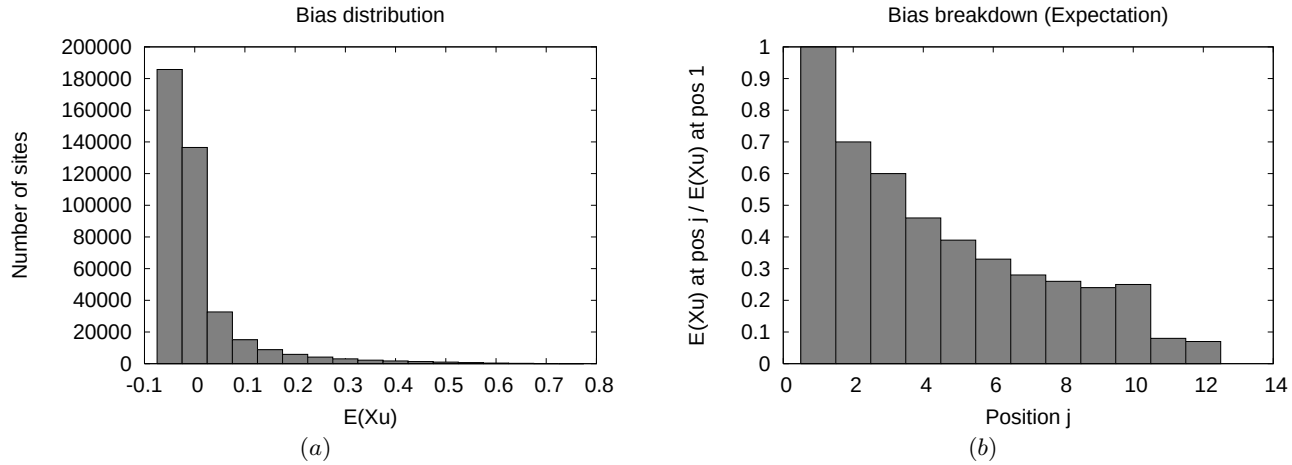

Figure 1: a) Bias distribution, and b) Bias breakdown over positions

	insertions		del./sub.	
	mean	var	mean	var
MT(3) vs MT(2)	0.66	0.62	0.10	0.11
MT(5) vs MT(3)	0.73	0.70	0.28	0.29
MT(10) vs MT(5)	0.38	0.85	1.10	1.28

Table 4: Portfolio changes

4.1 Setup

We implemented our search optimization algorithm, as discussed in Sec. 2.1, in a tool called *Diver* that offered functionalities for searching and diversifying the results. For solving the quadratic programming optimization in our algorithm, we used Gertz's and Wright's OOQP [9] and it turned out to be fast (see Sec. 4.5 for details). Diver used Google search, and for each query, up to 500 top results from Google were retrieved and reranked to improve diversity. Correlations between pages were estimated based on textual features as discussed in Section 3 and the relevance statistics were estimated as in Eq. 8. X_u was calculated at the site level and was the same for all pages on the same site, except that its was weighted for each page according to its rank in the result (as shown in Figure 1-b). The parameter e in Eq. 2 was set to the mean of the best five expectations, i.e. MT(5), hence it was query- and result- specific (see Section 3.3 for details on the choice of e).

For evaluation, top five results from Diver were compared to those from Google in terms of result relevance and the number of different query aspects retrieved. For Diver, top five results included those that received the highest weights in the optimal portfolios derived using the optimization in Eq. 1; if there were less than five results in the portfolio, the rest of the results were selected from those outside the optimal portfolio but with the largest Google ranks.

4.2 Wikipedia disambiguation pages

To measure the improvement in the number of different query aspects retrieved, we selected 50 disambiguation pages from Wikipedia and used the titles of these pages as queries in both Diver and Google. The disambiguation pages were identified using the query `site:en.wikipedia.org` "`may refer to`" `disambiguation` at Google and after removing topics with either less than 3 or more than 15 different aspects. We ensured that the selected topics were real queries tried in the past by checking their frequencies in a small query log. All queries appeared at least once; some appeared more than 4000 times. Returned results from the two systems were passed to evaluators who were asked to assign each document in the result to its closest matching sub-topic in Wikipedia; when there was no matching sub-topic, the evaluators could create their own. Finally the systems were assessed based on the number of different aspects retrieved in top r. Figure 2 shows this result with r varying from 1 to 5. Google does a relatively good job retrieving multiple query aspects and Diver slightly improves upon Google.

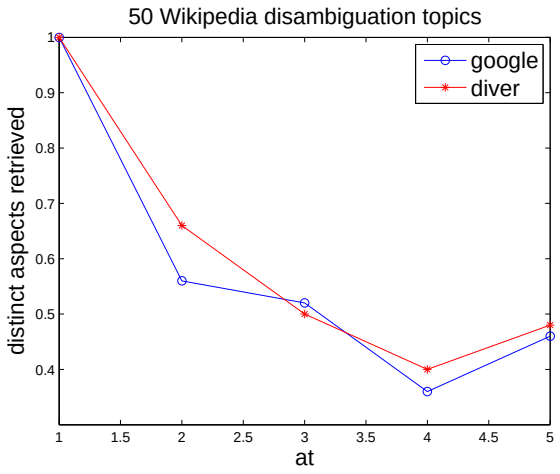

Figure 2: Unique aspects retrieved for Wikipedia queries

We found limitations in using the disambiguation pages to evaluate diversity. First, there are many obsolete aspects listed for topics that cannot be found elsewhere on the Web. On the other hand, more new usages of the topics often are not listed in Wikipedia. A good example of this is person

Figure 3: Unique aspects retrieved for random queries

names. A search on the Web often retrieves many unique persons with the same name but only a handful of them can be found in Wikipedia. Second, Wikipedia pages and topics are well-indexed by search engines and are not good representative queries. Finally, a manual assessment does not easily scale up to a large number of queries.

4.3 ODP categories

Our next experiment was on a randomly selected sample of queries from Google query log. After running the queries in both Diver and Google, we looked up the returned documents in Google directory and tagged them with the category names under which each document was listed. Two documents were treated in the same subject if they had at least one category in common. Figure 3 shows the mean number of unique aspects retrieved at r for $r = 1, \ldots, 5$, measured as the ratio of the number of different categories discovered and r. In 40% of the cases, the first document was not tagged, leading to a ratio of 0.6 at $r = 1$. Unlike the previous experiment, Diver improves upon Google in terms of the number of different aspects retrieved by a significant margin.

4.4 Result relevance

The previous experiments confirm our claim that the result diversity is improved in Diver. However, diversity is meaningful only if the retrieved results are relevant. To test for relevance, we selected from our query log a random sample of queries that were expected to have multiple aspects. The selection criteria was (a) to include queries that had less than 4 terms and a minimum unigram log frequency greater than 3 (i.e. every term in the query appeared in at least 1000 documents), as obtained from Web 1TB 5-gram dataset [4], and (b) to exclude queries that had either a number or one of the terms *weather, picture, map, yahoo, wikipedia* and *youtube* or returned 10 or less results at Google. This was based on the observation that both long queries and those that match very few documents or very specific sites are less likely to benefit from diversification. Also to alleviate the assessment process, we excluded queries that had all frequent terms (i.e. a minimum unigram log frequency greater than 5). With this criteria, we selected 42 queries out of 427 random queries we examined (i.e. 10% selectivity).

As for comparison with our system, Google established one baseline for us; since our system was built on top of Google, it was important to make sure that we are improving and not worsening the results. As another baseline, we chose Carbonell's and Goldstein's MMR [5] which combines query relevance with result novelty. The model, referred to as Maximal Marginal Relevance, ranks documents for a given query Q based on both their similarities to the query and also their dissimilarities to other selected documents, i.e.

$$MMR \stackrel{def}{=} \arg \max_{d_i \in R \setminus S} \left[\lambda(Sim_1(d_i, Q) - (1-\lambda) \max_{d_j \in S} Sim_2(d_i, d_j)) \right]. \quad (9)$$

where R is a set of documents retrieved for Q, $S \subseteq R$ is the set of documents selected already, $\lambda \in [0, 1]$ and was set in our experiments to 0.4 based on authors recommendation, and Sim_1 and Sim_2 are two similarity functions. A problem in implementing MMR is the choice of a similarity measure between documents and query; a textual similarity alone is not a good measure of relevance in the context of the Web. To overcome this problem, we used the reciprocal rank of d_i in the Google result for Q as our $Sim_1(d_i, Q)$ function. $Sim_2(d_i, d_j)$ was the standard Cosine similarity between feature vectors of documents, and each feature vector included the same set of features extracted within Diver (as explained in Sec. 3). Because of these changes, we refer to this version of MMR as MMR*. MMR* turned out performing consistently the best in terms of the precision of the results (as discussed next) among the variations we tried (e.g. compared to the case when $Sim_1(d_i, Q)$ was set to 1 for top k documents from Google for Q and zero for the rest).

The selected queries were submitted to Diver, Google, and MMR* and the top 5 results for each system were selected for user evaluation. The evaluators were asked to assign a score of 2 for relevant pages with enough new content, 1 for relevant pages with little new content, 0.5 for relevant pages with no new content and 0 for non-relevant pages. With this setting, a system that returned 5 relevant results all covering the same topic but with little variation would only get an average score of $(2 + 1 + 1 + 1 + 1)/5 = 1.2$ or less, whereas a system that returned relevant pages with more variations or differences was likely to score higher. This scoring quantifies a combined measure of both relevance and novelty very similar to α-nDCG metric of Clarke et al. [7], where a positive value of α indicates that novelty is rewarded in the results proportional to α. Compared to α-nDCG, our scoring is more discrete, making the assessment a bit easier for our evaluators.

For evaluation, we took some extra steps to make sure that there was no bias against one system. In particular, the ordering of the systems were randomized and varied from one query to next, and the evaluators didn't know which system was being shown first or last. Each query was assessed by two evaluators, and the score of a system on a query was the average score assigned to its results.

Table 5 shows the score (calculated as the ratio of average score and the maximum score) recorded for each system, averaged over 42 queries. Diver performs better than Google and the difference is significant (at $\alpha = 0.05$). Diver also outperforms MMR* and the difference is significant (at $\alpha = 0.01$). The difference between Google and MMR* is

Systems	Google	Diver	MMR*
Score	0.56	0.61	0.54

Table 5: The ratio of average score and maximum score at 5

not statistically significant. Checking the queries, we could see that in 60% of the cases Diver gave better results than Google, in 19% of the cases they scored the same, and only in 21% of the cases Google did better. Compared to MMR*, Diver did better in 50%, the same in 29% and slightly worse in 21%.

To get a better feeling for the kind of results returned by Diver, Table 6 shows the results of Diver, Google and MMR* for a few queries. These queries are not from our random query set and were posed to our system after an internal presentation of the work.

4.5 Optimization efficiency

The major overhead in diversifying search results, as suggested in this paper, is solving a quadratic optimization function. To assess this overhead, we varied the number of documents being passed to the optimizer and measured the running time. As shown in Figure 4, the running time increases with the number of documents, but even at an input size of 500 documents, the running time was under 1 second on a modest Pentium 4 dual-core PC. On the other hand, the input size is not expected to be large; documents with low ranks in the search engine ranking generally have low expectations. These documents don't have much chance of making to an optimal portfolio anyway, hence they may be dropped with no or very little affect on the optimization. Figure 5 shows the probability that a document returned at rank i (in our case by Google) makes to an optimal portfolio.

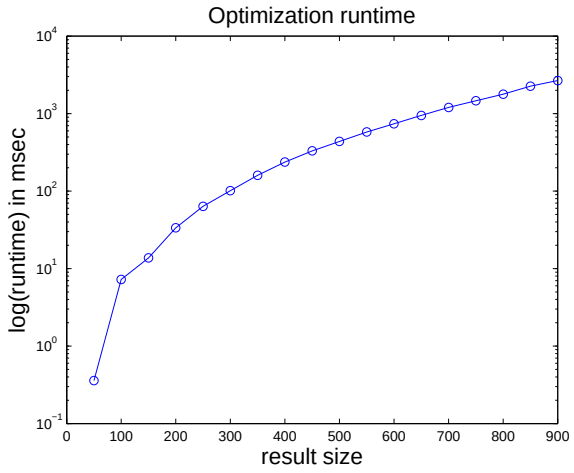

Figure 4: Optimization runtime

5. RELATED WORKS

Related work can be divided into the works on search diversification and portfolio optimization.

5.1 Search diversification

Carbonell and Goldstein evaluate MMR in document reordering and document summarization and report that more users prefer diversified results [5]. In a cost function similar to MMR, Zhai, Cohen and Lafferty [20] also combine novelty and relevance and use it for subtopic retrieval. A challenge in both works is finding a balance between relevance and novelty and adapting the functions to the settings of a search engine; since the computation of relevance scores within search engines may take into account many features including similarity, freshness, and even novelty, this double-dependence of ranks to novelty can affect the formulation and the results. In a method similar to MMR, Zhang et al. [22] use an affinity graph between documents where in each step a document with the largest score is selected while other documents with an affinity relationship to the selected document are penalized.

Chen and Karger [6] propose a Bayesian model with an objective function that puts at least one relevant document in top 10. To avoid evaluating the function for all subsets of size 10, the authors develop a greedy approach that selects one document at a time and does not change the previous selections as the algorithm progresses. Their greedy algorithm (similar to MMR) in each step selects a document that maximizes the objective function, conditional on the assumption that none of the previous selections were relevant. The model can be generalized to select at least k relevant documents in top 10 where k is fixed. Agrawal et al. [1] map queries and documents to ODP categories and propose an objective function that maximizes the probability that some document from each one of the categories a given query is assigned to is returned, conditional on the number of returned documents being fixed at some constant. The problem, as formulated, is NP-hard and the authors provide a greedy approximation to their formulation. Our approach does not fix k; in the context of the Web search, setting k can be challenging as it would require estimating the number of query aspects in advance. Also many queries (and sometimes documents) cannot be found in the ODP categories, as was the case for many person name and location queries we tried; for one location query in particular, Google reported 33,400 matches but the location was not in ODP. We adopt a numerical approach to the problem which easily scales to large result sizes. The recent addition of a diversity task to the Web track at TREC [10] is also related to our work and emphasizes the importance of the task.

Result diversification in general relates to the problem of document clustering, in that a diversified result set may be produced by combining the relevance scores with cluster information about each document. However, running clustering for diversification is an overkill; also clustering ranked documents and setting some of the parameters is not straightforward. As a client-side solution, Radlinski and Dumais [14] find for each query k other interesting queries in the log within 30min window of the query and merge top results of these queries to construct a more personalized but diversified result set.

The issue of diversifying search results over more structured data has also garnered some interest lately. Vee et al. [17] define a diversity ordering of attributes and a similarity measure that weights higher order attributes more heavily than attributes of lower order. As yet in one more domain, Zwol et al. [23] propose a method for diversifying image searching by sampling additional query terms from image

tags and further using those terms to find related but more diverse collection of results.

5.2 Portfolio optimization

Our portfolio model of search is based on Markowitz's Nobel-prize winning portfolio selection [11]. In his seminal work, Markowitz notes the relationship between expected return and the associated risk in the stock market and develops an optimization model that can minimize risk for a given level of return or vice versa. The model has been used in the areas outside finance, but to the best of our knowledge, not much work is done to incorporate a similar notion of risk to the Web search. With the exception of a recent independent work by Wang and Zhu [18] which studies some properties of the model when applied to a ranked list in IR, our work is the first that applies the model to diversify Web search results.

As in Markowitz's portfolio selection, risk manifests in Web search in the form of returning a result set which may not include a user's desired aspect. This can be due to ambiguity in query interpretations, uncertainty about users' intentions and sometimes heuristics that may be applied within a search engine. Our formulation of diversity tries to reduce risk by taking into account correlations between pages and that if a document is not relevant, other correlated documents are also likely to be not relevant.

6. CONCLUSIONS

We have proposed a model and an algorithm for diversifying search engine results, and have presented an evaluation and analysis of our algorithm and its results. To the best of our knowledge, this is the first work that relates results quality and diversity to expected payoff and risk in clicks and provides a model to optimize these quantities. A challenge in any search optimization including ours is deriving statistics about variables used in the model; we have presented a few methods to derive these statistics based on data and statistics that is generally available in search engines. There is room to find better statistics about click-through rates and correlations which can lead to more accurate estimates and better search results. Another interesting related direction is detecting queries that can or cannot benefit from a result diversification. We tried to do a little bit of this with our heuristics in Section 4, but this area by itself, to the best of our knowledge, is open for further research.

Acknowledgments

The authors would like to thank the members of NextRank group at Google for their input and the anonymous reviewers of the paper for their comments. This research was partially supported by the Natural Sciences and Engineering Research Council.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Leong. Diversifying search results. In *Proc. of ACM Conf. on Web Search and Data Mining*, 2009.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] R. Bhatia. *Positive Definite Matrices*. Princeton University Press, 2006.

[4] T. Brants and A. Franz. Web 1t 5-gram version 1. Linguistic Data Consortium, Philadelphia, 2006.

[5] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR Posters*, 1998.

[6] H. Chen and D. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proc. of SIGIR Conf.*, 2006.

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

[8] R. Fletcher. *Practical methods of optimization*. Wiley and Sons, second edition, 1987.

[9] M. Gertz and S. Wright. Object-oriented software for quadratic programming (ooqp). http://pages.cs.wisc/edu/ swright/ooqp.

[10] H. Craswell, C. Clarke, I. Soboroff. TREC 2009 novelty track. In *Proc. of TREC*, 2009.

[11] H. Markowitz. Portfolio selection. *The Journal of Finance*, VII(1):77–91, 1952.

[12] J. Nocedal and S. Wright. *Numerical optimization*. Springer, second edition, 2006.

[13] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *The 1st Intl. Conf. on Scalable Information Systems*, 2006.

[14] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proc. of SIGIR Conf. (Poster Session)*, 2006.

[15] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proc. of WWW Conf.*, pages 521–529, 2007.

[16] J. Teevan, E. Adar, R. Jones, and M. Potts. Information re-retrieval: repeat queries in yahoos logs. In *Proc. of SIGIR Conf.*, pages 151–158, 2007.

[17] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *Proc. of the ICDE Conf.*, pages 228–236, 2008.

[18] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *Proc. of SIGIR Conf.*, pages 115–122, 2009.

[19] Wikipedia. http://en.wikipedia.org.

[20] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proc. of SIGIR Conf.*, 2003.

[21] C. Zhai and J. Lafferty. A risk minimization framework for information retrieval. In *Proc. of SIGIR Workshop on Mathematical/Formal Methods in IR*, 2003.

[22] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W. Ma. Improving web search results using affinity graph. In *Proc. of SIGIR Conf.*, 2005.

[23] R. Zwol, V. Murdock, L. Pueyo, and G. Ramirez. Diversifying image search with user generated content. In *Proc. of the 1st ACM Conf. on Multimedia IR*, pages 67–74, 2008.

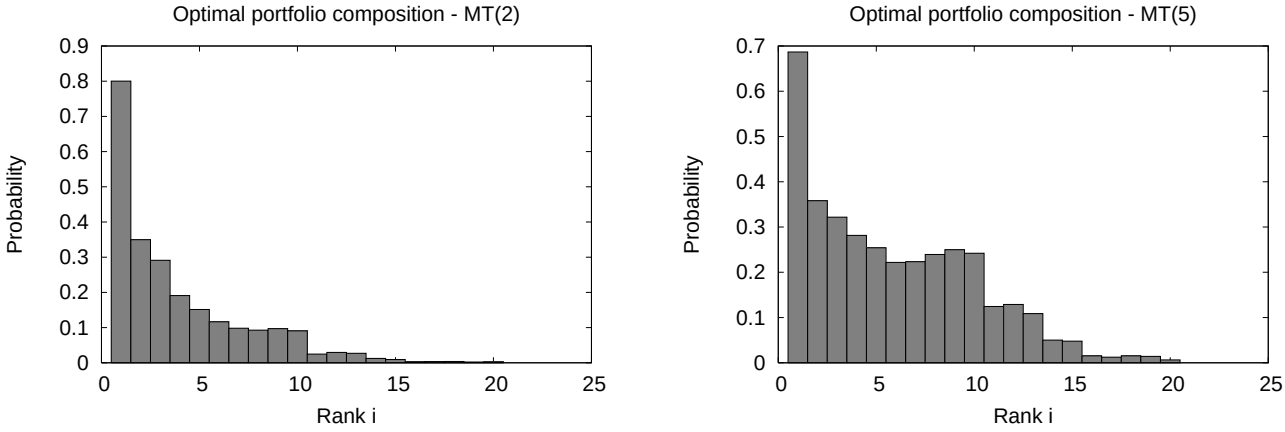

Figure 5: Probability that a document at rank i makes to an optimal portfolio

Query: *manber*			
	Google	Diver	MMR*
1	Udi Manber - old home page	Udi Manber - Wikipedia	Udi Manber - old home page
2	Udi Manber - Wikipedia	Jeffrey Manber - Wikipedia	David Manber - imdb.com
3	Udi Manber - publications	Rachel Manber - academic profile	Udi Manber - Wikipedia
Query: *sergey*			
	Google	Diver	MMR*
1	Sergey Brin - Wikipedia	Sergey Brin - Google Management	Sergey Brin - Wikipedia
2	Sergey Brin - Google Management	Sergey Korolyov - Wikipedia	Sergey Brin - Stanford
3	Sergey Brin - Stanford	Sergey Formin (at U. Mich.)	Sergey Brin (at forbe.com)
Query: *hilton*			
	Google	Diver	MMR*
1	Hilton hotel	Hilton hotel (HHonors)	Hilton hotel
2	Hilton hotel online	Perez Hilton blog	Hilton hotel (at Germany)
3	Hilton hotel (HHonors)	Hilton Vacations Club	Hilton hotel online
Query: *bush*			
	Google	Diver	MMR*
1	George W. Bush - Wikipedia	George W. Bush (at whitehouse.gov)	George W. Bush - Wikipedia
2	President G.W. Bush (at whitehouse.gov)	George Bush's library (in Texas)	President G.W. Bush (at whitehouse.gov)
3	President of US (at whitehouse.gov)	George W. Bush - Wikipedia	Jibjab - funny jokes
Query: *jaguar*			
	Google	Diver	MMR*
1	jaguar.com (car)	jaguar.com (car)	jaguar.com (car)
2	jaguarusa.com (car)	jaguarusa.com (car)	schrodinger.com (not related)
3	jaguar-Wikipedia (animal)	jaguar-Wikipedia (animal)	jaguar.ca (car)
Query: *python*			
	Google	Diver	MMR*
1	python.org (prog. lang.)	Python - Wikipedia (prog. lang.)	python.org (prog. lang.)
2	Python - Wikipedia (prog. lang.)	Monty python - Wikipedia	Python - Wikipedia (prog. lang.)
3	python.org/download (prog. lang.)	python.org (prog. lang.)	python.org/download (prog. lang.)

Table 6: Example results from Diver, Google and MMR* for various queries