Appeared in CIKM 2016. Paper available at http://bit.ly/measuringMetrics

Measuring Metrics

Pavel Dmitriev
Microsoft Corporation
555 110th Ave NE
Bellevue WA 98004, USA
+1-425-4211185
padmitri@microsoft.com

Xian Wu
Microsoft Corporation
5 Danleng Street, Haidian
Beijing 100080, China
+86-10-59172836
xianwu@microsoft.com

ABSTRACT
You get what you measure, and you can't manage what you don't measure. Metrics are a powerful tool used in organizations to set goals, decide which new products and features should be released to customers, which new tests and experiments should be conducted, and how resources should be allocated. To a large extent, metrics drive the direction of an organization, and getting metrics "right" is one of the most important and difficult problems an organization needs to solve. However, creating good metrics that capture long-term company goals is difficult. They try to capture abstract concepts such as *success*, *delight*, *loyalty*, *engagement*, *life-time value*, etc. How can one determine that a metric is a good one? Or, that one metric is better than another? In other words, how do we measure the quality of metrics? Can the evaluation process be automated so that anyone with an idea of a new metric can quickly evaluate it? In this paper we describe the metric evaluation system deployed at Bing, where we have been working on designing and improving metrics for over five years. We believe that by applying a data driven approach to metric evaluation we have been able to substantially improve our metrics and, as a result, ship better features and improve search experience for Bing's users.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]

General Terms
Experimentation

Keywords
Measurement, Quality, Search Metrics, Online Experimentation, A/B Testing.

1. INTRODUCTION
Metrics are a powerful tool used in organizations to set goals, decide which new products and features should be released to customers, which new tests and experiments should be conducted, and how resources should be allocated. It is common to set performance goals for individual teams based their contribution to a key metric. For example, for a search engine such as Bing or Google a key metric may be NDCG [11]. Each team responsible for a specific component of the search engine - crawling, indexing, ranking, spelling correction, etc. - is expected to contribute a certain amount to the overall improvement on NDCG.

Correctly chosen metrics incentivize teams to take actions that are in the long-term interest of the company, while poorly chosen metrics may lead to counterproductive decisions and actions [15]. In this sense metrics drive the direction of an organization, and getting metrics "right" is one of the most important and difficult problems an organization needs to solve.

Creating good metrics is difficult. Metrics often try to capture abstract and subjective concepts such as *success*, *delight*, *loyalty*, *engagement*, *life-time value*, etc. These concepts represent real organizational goals for serving their customers, but there's no standard way to formally define them.

Consider the following example. A search engine may want to define a "success" metric to measure how often its users are successful in finding the information they need [7]. A simple way to define success for a user issuing a query q is based on the time the user spent on a result after clicking on it [14], such as

$$success(q) = \begin{cases} 1, if\ q\ had\ a\ click\ with\ dwell\ time > 30\ seconds \\ 0, otherwise \end{cases}$$

Although this definition was found to fit many scenarios [14], it still has many issues. It does not capture "good abandonment" [20] scenario where users get the answer to the query directly from search engine's result page without clicking on anything (e.g., "time in Rome Italy"). It also does not capture exploratory scenario where users browse quickly through search engine results pages, not necessarily looking for a single answer (e.g., "new movies"). It does not specify how to interpret clicks on a "related search" query suggestion, or how to interpret clicks that go to the search engine's "vertical" experiences such as images or videos rather than to an external result page. If this simple success metric is set as an organization's goal, all of the above scenarios would suffer (e.g., adding a new good abandonment feature to the page may cause a decrease in long dwell time clicks, thus degrading our success metric). Creating a good "success" metric that captures all of the above scenarios is a difficult, iterative process and is an active area of research within search community (see Jiang et. al. [12] for one of the recent works in this area).

We believe that the key to success in designing good metrics is the ability to measure the metric's quality, and the ability to compare metrics (e.g. a new version of "success" metric vs the old version). In this paper we share our experience in applying such data driven approach to evaluating metrics in the context of Bing search engine. Our main contributions are as follows:

CIKM'16 , October 24 - 28, 2016, Indianapolis, IN, USA

ACM 978-1-4503-4073-1/16/10..$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983356

- We define important characteristics of good metrics and describe how they are formalized as meta-metrics used in Bing to evaluate metric quality
- We describe the system architecture for evaluating quality of metrics, addressing data quality, scalability, and performance issues.
- We apply the framework to several common search metrics, obtaining insights in their behavior and showing how to obtain better, more sensitive metrics
- Finally, we give three real-world examples of applying the above framework to evaluate metric improvement ideas

The system we describe is a production system that has been used in Bing to evaluate and inform the design of new metrics for the last several years. While the discussion in this paper is scoped to measurement problems arising in a search engine, we believe the principles and, to a large extent, system architecture will apply more generally to a wide range of domains.

The paper is organized as follows. After reviewing the concept of controlled experiments (Section 2) and related work (Section 3), we describe our metric evaluation framework in Section 4. Section 4.1 discusses how to build an experiment corpus for evaluation, Section 4.2 discusses the system architecture and how it is deployed in the real world, and Section 4.3 proposes several meta-metrics to evaluate metric quality and shows how some commonly used search metrics compare based on these criteria. Section 5 gives three examples of metric design questions that we were able to answer using our framework. Section 6 summarizes the impact our system had in Bing, and Section 7 concludes.

2. CONTROLLED EXPERIMENTS

In the simplest controlled experiment or A/B test users are randomly assigned to one of the two variants: control (A) or treatment (B). Usually control is the existing system, and treatment is the existing system with a new feature X added. User interactions with the system are instrumented, and metrics are computed. If the experiment was designed and executed correctly, the only thing consistently different between the two variants is the feature X. External factors such as seasonality, impact of other feature launches, competitor moves, etc. are distributed evenly between control and treatment and therefore do not impact the results of the experiment. Therefore any difference in metrics between the two groups must be due to the feature X. This establishes a causal relationship between the change made to the product and changes in user behavior. For a survey of controlled experiments on the web see Kohavi et. al. [16].

While metrics are very important in contexts other than controlled experiments (reporting/dashboards, cohort studies, pre-post analyses, etc.), evaluating metrics in the context of controlled experiments allows focusing evaluation on how metrics respond to actual product changes, rather than on metrics' sensitivity to external factors. Because of this, in this paper we discuss the measurement problem in the context of controlled experiments.

3. RELATED WORK

Related work can be grouped into three categories: controlled experiments, individual metric improvements, and principles of metric design.

Controlled experiments is an active research area, fueled by the relative ease with which a large number of users can be reached on the web [16]. Research focused on scalability of experimentation systems [18, 26], new statistical methods to improve sensitivity [5], rules of thumbs and lessons learned from running controlled experiments in practical settings [19, 17], and projections of results from short-term experiment to the long term [10]. Applying new statistical methods, such as the variance reduction technique described in Deng et. al. [5], makes the existing metric more sensitive. To our knowledge, our work is the first to focus on comparing and evaluating different metrics in the context of controlled experiments. Our approach can also be used to evaluate sensitivity improvements from techniques like [5] in terms of impact on the real decisions made using the metric.

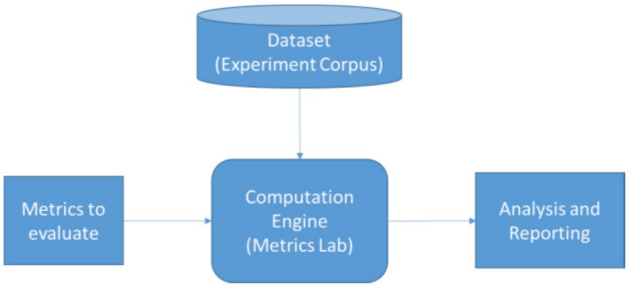

Figure 1: Metric Evaluation Framework

A large number of works focused on improving an individual metric: NDCG [21] or user satisfaction [8] in the area of learning to rank, MAP or AUC in the area of collaborative filtering [27], etc. Our work is different in that we propose a metric evaluation approach to inform the choice of the metric to optimize. This is extremely important since, as in the "success" metric example from section 1, optimizing a poorly chosen metric may result in hurting company's goals.

The importance of metrics as a mechanism to encourage the right behavior has been recognized and studied in the management domain since at least 1956 [23]. It is articulated using statements such as "What gets measured, gets managed" [25], "What you measure is what you get" [13], and "You are what you measure" [9]. A good survey of research can be found in Blackburn et. al. [2]. The topic of finding good organizational metrics is also discussed in several books [1, 3, 6, 22]. These works focus more on the organizational challenges in identifying, selecting, and implementing good metrics. Recently, we shared several lessons on defining metrics for controlled experiments in [4], where defining "metrics for metrics", or meta-metrics, was called out as the key need. In this paper we discuss how such meta-metrics can be defined, and describe the system used in Bing for evaluating metric quality.

4. METRIC EVALUATION FRAMEWORK

Figure 1 lists the main components of our metric evaluation framework. The most important component of the framework is a dataset of historical controlled experiments, *Experiment Corpus*. In Bing we use several different corpuses: a corpus of randomly selected recent experiments; a manually curated corpus containing important and representative experiments from different feature areas labeled by human experts as positive or negative; small corpuses created by specific feature teams (e.g., Ads team has their own corpus of experiments related to tuning ad selection, placement, and look and feel). We discuss the construction and properties of experiment corpuses we use in section 4.1.

The input to the system is the metrics that need to be evaluated. Typically, this includes new metrics that the user came up with, and some existing metrics that the user wants to use as baselines. New metrics need to be formally defined using a specified syntax, while definitions of existing metrics are already present in the system.

We call our computation engine *Metrics Lab*. The job of Metrics Lab is to efficiently compute statistics for each metric on each experiment in the Experiment Corpus. The challenge here is that the direct approach of going over the search engine logs, extracting the experiment data, and computing the metrics is prohibitively expensive. We describe Metrics Lab and performance optimization techniques it employs in section 4.2.

Once the statistics for each metric are computed, we evaluate each metric according to several quality criteria (meta-metrics) and generate reports for individual performance of each metric as well as for metric comparisons. Metric evaluation criteria are discussed in section 4.3.

4.1 Experiment Corpus

Experiment corpus used for evaluation has major impact on the trustworthiness of results. Any biases or quality issues with the corpus will affect the results and may lead to wrong conclusions. Over the years, we refined the ways to construct the corpus, ending up with several corpuses for different types of evaluation.

The first corpus is a randomly selected sample of recently run experiments. This sample is representative of the experiments currently run in the company, and metric evaluation results based on this corpus should directly generalize to all experiments. Obtaining this corpus, however, is not as simple as it sounds. Taking a simple random sample of experiments from the system brings in many *incorrectly configured*, *untriggered*, and *underpowered* experiments. Evaluating metrics on such experiments may lead to wrong conclusions. We discuss the issues such experiments present and how to detect them below.

Incorrectly configured experiments have an error in experiment configuration. For example, experiment owner may configure the treatment to run only on a certain browser, but forget to restrict the control to only this browser. This results in more users included in control and in an unfair comparison. Including such an experiment in evaluation is meaningless and may create bias. Experiments like the one above can be detected by comparing the expected fraction of users in the experiment to the actual, and filtering out the experiments where the difference is statistically significant. Sometimes the issues are subtler such as incorrect instrumentation, duplication of calls, etc. We use a number of filters based on experiment metrics being within a "reasonable" range to filter out such cases. For example, if an experiment doubles the revenue it is "too good to be true" and is almost certainly the result of a misconfiguration.

Many features evaluated in experiments apply to only a small fraction of queries. For example, an experiment may be changing the look and feel of the calculator answer that shows up for queries such as "4200 / 75". Because few users issue such queries, to detect changes in metrics the experiment needs to have sufficient power (practically, enough users who issued such queries), and the correct triggering logic to limit the analysis only to the users who issued such a query at least once. If such an experiment is underpowered (not enough users) or untriggered (triggering logic is not provided), the signal from the feature will drown in a lot of noise from users and queries not affected by the feature, resulting in no statistically significant changes on the metrics of interest. Including such an experiment in evaluation is a waste of computational resources as none of the metrics of interest are likely to show statistically significant movements. One way to detect such experiments is by testing whether the fraction of statistically significant metrics observed in the experiment is greater than expected by chance, and only include experiments where it is the case.

Our second corpus is manually constructed based on experiment "interestingness" for metric evaluation. Interesting experiments are representative experiments from different feature areas, "learning" experiments that were run for the sole purpose of understanding user behavior, experiments that had known bugs negatively impacting users, etc. An important category of interesting experiments are the experiments where the current metrics do not work. For example, if success metric from section 1 is our main metric, then a "good abandonment" experiment that we believe is good for users but regresses the success metric is interesting and is worth including in the corpus.

We label each experiment in the "interesting" corpus as positive or negative with respect to user value – did users have better experience with Bing because of this feature? To obtain these ground truth labels, one may be tempted to simply check if the experiment was eventually shipped to production. However, at least in Bing this is not a reliable indicator. Many experiments are run for the sole purpose of understanding user behavior and are not intended to be shipped. Experiments that aren't shipped are often iterations on a positive idea, just not the final shipped version of it. Many experiments are infrastructural changes that are shipped but do not have user impact. Finally, some experiments are shipped because they are stepping stones for something bigger and by themselves aren't necessarily positive.

Because of this we employ a manual process for obtaining ground truth labels. One has to be careful to avoid assigning the label based only on the existing metrics – this would bias the labels to favor the existing metrics that we want to improve upon. We review each candidate experiment with the experiment owner and a panel of experiment analysis experts, looking not only at the existing metrics, but also at user studies that were done on the feature, user feedback relevant to the feature, and any other available data to make the most accurate decision possible on whether the experiment is good or bad for users. This is a slow and expensive process, but it produces high quality labels.

It is important to note that, while the label and the decision whether to include the experiment in the corpus are not based solely on the existing metrics, the process of constructing the "interesting" corpus is still inherently biased. One needs to be careful when trying to generalize the results from this corpus to all experiments. Because of this, we usually encourage analysts to test the results obtained from this corpus on the randomly sampled corpus as well.

On the positive side, however, we found that the "interesting" corpus is usually better at highlighting the differences between metrics, compared to a randomly sampled corpus of the same size. The existence of ground truth labels allows evaluating metrics with respect to the true user value, as opposed to just measuring their sensitivity. Experiments in the "interesting" corpus are usually well documented, allowing the analyst do qualitative analysis to determine what caused a change in the metric (or lack of thereof) in a specific experiment, and form hypotheses on how to further improve the metric. Due to these benefits, analysts usually start their evaluation on the "interesting" corpus, and after obtaining a

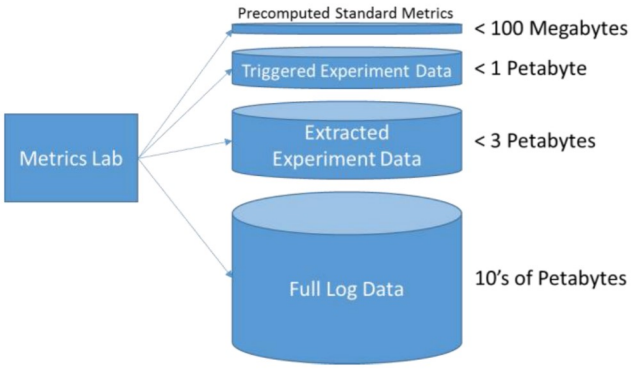

Figure 2. Cache structure for Metrics Lab

metric candidate that worked well on the interesting corpus, evaluate it on a randomly sampled corpus[1].

Some feature teams also maintain small corpuses for their areas. For example, ads team has their own corpus of experiments related to tuning ad selection, placement, and look and feel. They use these corpuses for improving their feature-specific metrics.

4.2 Metrics Lab

As noted above, direct metric evaluation over even a moderate size Experiment Corpus is very expensive due to huge volume of data that would need to be processed. Suppose, for example, that Experiment Corpus has 100 experiments and each experiment is 2 weeks long. Naively extracting the data for these experiments one by one from raw search engine logs would require reading close to 14*100/365 ≈ 4 years' worth of log data, dozens of petabytes (compressed) in the case of Bing.

Metrics Lab solves this computational problem via a tiered cache shown in figure 2. It automatically optimizes metric evaluation jobs by choosing the highest cache tier possible. The performance gains from introducing the cache were as follows. For a 100-experiment corpus, processing an average metric evaluation job using only Full Log Data would require ~100 hours or over 4 days. Extracted Experiment Data cache stores only the data for users in the experiment. Running on extracted data instead of the full log reduces the running time by an order of magnitude to ~10 hours. Most evaluation jobs need only the triggered user population. Introducing Triggered Experiment Data cache further reduces the running time to ~7 hours. Finally using precomputed standard metrics rather than re-computing them on the fly further reduces the running time to ~5 hours. The running time reduction here is due to some standard metrics being rather complex requiring more than 1 pass over the data.

Figure 3 depicts the process of setting up and running a metric evaluation job from the analyst's point of view. The process is simple enough so that even non-technical people are able to use it to evaluate their metric ideas.

The analyst starts by creating a profile for his or her job, specifying the names of the metrics to be evaluated, the subset of the experiment corpus to run on, as well as other operational parameters.

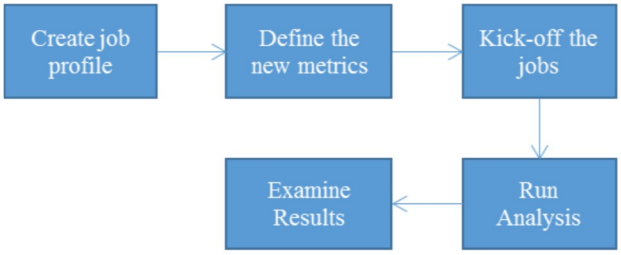

Figure 3: The analyst's workflow with Metrics Lab

Next the analyst uses an SQL-like language to define metric computation logic for the new metrics. Most of the time this is done by copying and editing the definition of a similar standard metric, but the language is powerful enough to allow very complex metric definitions.

Next the analyst kicks off the job. Metrics Lab will automatically determine the cache tiers that need to be accessed and will generate and submit a series of scripts to the cloud cluster where the data is stored.

Once the scripts are completed, the analyst kicks-off the analysis script which downloads and merges the results, calculates evaluation criteria, and generates reports.

4.3 Metric Evaluation

In [9] Hauser and Katz listed two important properties of a good metric: (1) improving the metric should move the company towards its long-term desired outcomes, and (2) individual teams should be able to directly impact the metric. Both of these properties are important. A metric that does not correlate with the company's goals will result in development efforts focused in a wrong place, and a good metric that the team does not have ability to directly impact is not useful for directing that team's work. In this section we introduce several meta-metrics (metrics for comparing other metrics) that will measure these properties. We kept the definitions of meta-metrics simple, to allow for easy interpretability and debug-ability.

4.3.1 Sensitivity

In the context of a controlled experiment, sensitivity of a metric refers to the amount of data needed for the metric to show that a treatment-control delta of a specific magnitude is statistically significant. Sensitivity is important because more sensitive metrics allow detecting small changes sooner, shortening the time required for running an experiment and improving experimentation and decision making agility. While, as we discuss later, sensitivity is not the only aspect to consider when deciding which metric to focus on, comparing metrics on the sensitivity axis provides useful insights.

Assuming we are not changing the statistical test used, sensitivity depends on 3 factors: the amount of data (number of users or queries in our case), the variance of the metric, and the effect size (treatment-control delta). The first two factors are properties of the metric itself and do not depend on the type of the experiment that is run. The last one, however, directly depends on what kind of experiments are being run in an organization. A metric that is sensitive for one type of experiment may not be sensitive for

[1] Since it is expensive to compute a new metric on a very large number of historical experiments, this evaluation is often done by shipping the new metric to production as beta, waiting to accumulate enough experiments with this metric pre-computed, and then running the evaluation.

another type. For example, Page Load Time may be a very sensitive metric for experiments that involve adding or removing visual features on the page (because loading a visual feature takes time), but may not be a sensitive metric for experiments that involve changes to the ranking algorithm that executes on the backend at roughly constant time. Thus it is important to evaluate metric sensitivity in the context of the actual experiments run in the organization. For us, this context is defined by the Experiment Corpus.

Let m be a metric, and $\{e_1, e_2, \cdots, e_N\}$ the set of experiments in the corpus. Let t_i be the test statistic obtained by applying a statistical test to the metric m in the experiment e_i, and $abs(t_i)$ be its absolute value.

$$Sensitivity(m) = \frac{\sum_{i=1}^{N} abs(t_i)}{N}$$

Larger test statistic translates to a smaller p-value and a more sensitive metric. A larger *Sensitivity* score over the whole corpus means the metric is more sensitive over different types of experiments, or that more teams are able to impact the metric – one of the desired properties of a good metric.

We also define a *BinarySensitivity* metric which is more robust to outliers as well as easier to interpret. Let t be the statistical significance threshold used to decide whether a metric is statistically significant, and $I(a)$ be an indicator function that evaluates to 1 if a is true and to 0 otherwise. Then *BinarySensitivity* is simply a fraction of experiments in the corpus for which the metric was statistically significant.

$$BinarySensitivity(m) = \frac{\sum_{i=1}^{N} I(t_i < t)}{N}$$

Figure 4 shows the *BinarySensitivity* scores for some commonly used metrics, with t = 1.96 (corresponding to p-value of 0.05). Definitions of these metrics are given in Table 1.

There are several interesting observations in this comparison. Simply counting the number of events the user had, such as queries, sessions, or queries with clicks does not result in sensitive metrics. The three lowest sensitivity metrics in the table are in this category. To obtain more sensitive metrics one needs to normalize the counts. *Queries per Session* metric is more sensitive than *Queries per User*, and *Query Click Rate* is more sensitive than *Queries with Clicks per User*. In most cases this is due to variance reduction caused by normalization. While the overall number of queries the user issues may grow very large over the course of an experiment, the number of queries per session is more bounded resulting in increased sensitivity. Similarly, moving from counting queries with clicks to query click rate (queries with clicks / total queries) bounds the value of the metric to be between 0 and 1, resulting in even larger sensitivity gain. Another common technique for increasing the sensitivity of count metrics is truncation, which reduces the variance by eliminating outlier values. In [19] it was shown that capping Revenue per User metric at $10 allows detecting 30% smaller changes with the same number of users.

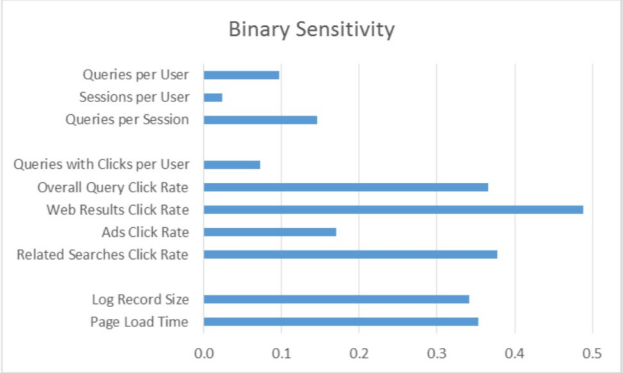

Figure 4. The Sensitivity of Common Search Metrics

Looking at the metric group in the middle of Figure 4, we can observe that, while the Overall Query Click Rate is a fairly sensitive metric, some of its components, especially Web Results Click Rate,

Table 1. Definitions of common search metrics.

Metric	Definition
Queries per User	Average number of queries issued by a user.
Sessions per User	Average number of sessions the user had. Session breaks are defined as 30-minute period of inactivity.
Queries per Session	Average number of queries in a session the user had.
Queries with Clicks per User	Average number of queries with clicks the user had.
Overall Query Click Rate	Average number of queries with clicks divided by the total number of queries the user had.
Web Results Click Rate	Average number of queries with clicks on Web Results divided by the total number of queries the user had.
Ads Click Rate	Average number of queries with clicks on Ads divided by the total number of queries the user had.
Related Searches Click Rate	Average number of queries with clicks on Related Searches divided by the total number of queries the user had.
Page Load Time	Average time between the user issuing a query and the page with results loaded in the browser.
Log Record Size	Average size of the log record for the queries user issued.
Time to Click	Average time from the start of a user session (first query), to the first result click.
Time to Long Click	Average time from the start of a user session (first query), to the first result click with dwell time of at least 30 seconds.

are more sensitive than the overall. This is because it is relatively easy in an experiment to shift clicks from one area of the page to another, but it's more difficult to increase the overall number of clicks [19].

This is also the reason why Related Search Click Rate metric shows high sensitivity even though there are very few experiments in the corpus that directly affect related searches: improvements to search quality reduce the need for query reformulation, while search quality degradations increase it, both affecting engagement with related searches. This highlights a common pitfall in analyzing online controlled experiments, where a team responsible for a certain feature on the page is tempted to call their experiment a success when the engagement to their feature increases. Most of the time this is due to cannibalizing the engagement from other areas on the page, not due to increasing the overall engagement.

The last group in Figure 4 shows two examples of system-level metrics. These metrics measure operation of the search engine rather than user behavior. They are usually quite sensitive, but, as one may expect and we show in the following section, they are not good predictors of user value.

4.3.2 Alignment with User Value

In Bing we consider customer satisfaction, or "user value," one of the key long-term objectives of the search engine. Happier users will lead to increased usage, share growth, and more advertising revenue. Recall that each experiment in the "interesting" corpus is labeled as "positive" or "negative" with respect to the user value, based not just on the observed metric changes but also on manual judgments, user feedback, and any other available data. These labels are used to evaluate the alignment of a metric with user value.

Let N_a^+ be the number of experiments in the corpus for which the metric *m* is statistically significant and the direction of the metric agrees with the label (treatment-control delta is positive and label is "positive," or delta is negative and label is "negative"), and let N_a^- be the number of experiments in the corpus for which the metric *m* is statistically significant and disagrees with the label. We experimented with several versions of *LabelAgreement* defined by the formula below.

$$LabelAgreement(m) = \frac{w_1 * MAX(N_a^+, N_a^-) - w_2 * MIN(N_a^+, N_a^-)}{N}$$

Here w_1 and w_2 are non-negative weights that sum up to 1. Since for some metrics smaller values are better (e.g. Page Load Time), *MAX* operator in the formula above will count the number of agreements, while *MIN* operator will count the number disagreements with the labels. By varying the weights one can place more emphasis on the labels the metric "got right," or more penalty on the labels it got wrong.

For the types of evaluations we have run, we found that agreement is more important than disagreement. Disagreement usually happens for known reasons and is highlighted by other metrics, warning the analyst. Because of that, we mostly use a simple version of *LabelAgreement* that only counts agreements:

$$LabelAgreement(m) = \frac{MAX(N_a^+, N_a^-)}{N}$$

Note that *LabelAgreement* incorporates both sensitivity (the metric is required to be statistically significant to count) and alignment with user value. Therefore it is used as a primary evaluation criteria on the "interesting" corpus. Figure 5 shows the *LabelAgreement* scores for the metrics we looked at above.

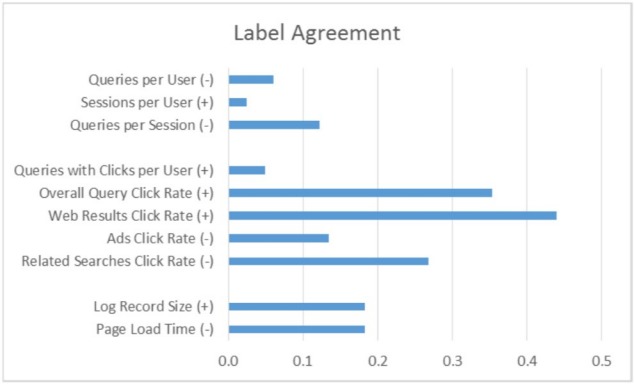

Figure 5. The Label Agreement of Common Search Metrics

Sessions per User metric has the lowest agreement score. Intuitively this metric has a great alignment with user value – if the user is coming to the site more often we must have done something good. For this reason, it is often used as a key site metric either by itself (also called *Site Visits*), or embedded in other metrics such as Daily Active Users or Monthly Active Users where "active" is defined as having one or more sessions. In practice, however, we found that this metric is very hard to improve in an experiment. Assuming that the user's need for web search is constant, a change in this metric effectively means gaining search share from a different search engine. It is hard to affect the behavior of a large enough fraction of users strongly enough to observe this effect during a short time span of an experiment. As discussed earlier, the fact that it is an "event count" metric also contribute to *Sessions per User*'s insensitivity.

We can see that Click Rate metrics have the highest LabelAgreement in this group, with *Web Results Click Rate* being the most sensitive metric. This is not surprising. Making changes that improve the quality of results and help users reach and evaluate web results easier are usually indicative of increased user value.

One may wonder if there are better metrics that Web Results Click Rate. The answer is "yes." Metrics with the highest LabelAgreement in Bing utilize the concept of *success* mentioned earlier. Due to the proprietary nature of the success definition used in Bing we do not discuss these metrics in this paper. Nevertheless, Web Results Click Rate is one of the most sensitive "simple" metrics.

As hypothesized earlier, system-level metrics, while quite sensitive, have fairly low LabelAgreement.

It is also interesting to observe the direction of LabelAgreement, indicated on Figure 5 by (+) or (-) next to the metric name. (+) means that increased metric values lead to higher agreement, while (-) means decreased metric values lead to higher agreement. *Ads Click Rate* has a negative direction, indicating that changes that increase user engagement with ads typically degrade user value and, conversely, degrading the quality of search results makes users engage with ads more. While one may think that making users issue more queries is better (contributes to higher "query share"), *Queries per User* and *Queries per Session* agree with labels better in the negative direction. The explanation for

this is that improving the quality of search results usually means that users do not have to reformulate their queries as much, leading to fewer queries per session and overall. On the other hand, degrading search quality or making search results hard to find and examine leads to users not finding what they want and issuing more reformulations.

4.3.3 Automation

To help the analyst, Metrics Lab generates automated reports with all of the meta-metrics described above. It also generates pairwise comparisons for every pair of metrics that analyst specified in the profile, listing the specific experiments where the two metrics disagree (e.g. one metric is statistically significant positive and the other one is statistically significant and negative or not statistically significant). Other debug information, such as metrics for the most frequently affected queries in the experiment, is generated as well. In practice, many insights into metric behavior are obtained from analyzing this debug information. Automatically analyzing such information and auto-generating insights from this analysis is one of the directions for our future work.

5. EXAMPLES

In this section we provide three examples of real metric improvement ideas, which we evaluated by applying the metric evaluation framework described above. Rather than reporting on the results in detail (which we cannot do due to proprietary nature of some of the metrics used), our intention here is to present representative questions the metric evaluation framework can be used to answer, that are hard to answer using traditional evaluation approaches based on manually labeled data.

5.1 Dedup or not Dedup?

"Duplicate queries" are the same query issued by the user twice in a row within the same session. Such queries are fairly common in search engine query log, accounting for close to 10% of all queries. Some of these queries are real user-initiated queries, while others could be due to lost browser cache, unintentional double-clicking, errors in calls from 3rd-party systems, etc.

An interesting question is whether it is better to use all queries for metric computation, or to first "dedup" by merging duplicate queries and taking a union of user actions (clicks, hovers, etc.) Intuitively, dedupping will eliminate noise from non-user-initiated duplicate queries, which should have positive effect on metrics. But it will also lose signal from real user-initiated duplicate queries which could hurt. This decision is important because it affects quantities such as number of queries and number of clicks per query that are part of many key search engine metrics.

Answering this question using the traditional approach based on collecting labeled data would be very difficult. It would require collecting labels to distinguish user-initiated and non-user-initiated duplicates. Even if this is accomplished, it's not clear how to correctly evaluate the tradeoff between noise reduction and loss of signal.

The evaluation using our methodology is straightforward. We picked three key metrics and implemented two versions of each metric, with and without dedupping.

Table 2. Impact of dedupping on metric quality.

Metric	Sensitivity	LabelAgreement	Label Disagreement
Query Click Rate	10	4	0
Query Long Click Rate	-2	0	-2
Quickback Rate	10	3	-3

The results are shown in Table 2. The table shows absolute deltas in Sensitivity, LabelAgreement, and LabelDisagreement. Comparisons where dedupped metric performs better are highlighted in green, and those where non-dedupped metric is better are red. We see that de-dupped versions of metrics perform better on most evaluation criteria, for most metrics.

The only red cell in the table presents an opportunity for a deep dive. Using the debug tools described above, we were able to quickly determine that, while the delta in Sensitivity was two experiments, there were total of nine experiments where the two metrics differed in their alignment. By examining those nine experiments in detail we were able to understand better the impact of dedupping and convince ourselves that it is indeed better to dedup the query stream before metric computation.

5.2 Metric Sensitivity to Threshold Changes

Many metrics rely on a threshold to determine their value. For example, it is a common practice [14] to use a threshold for click dwell time to determine whether the search engine result that was clicked on satisfied the user's information need.

We define dwell time of a click c to be the time user spent on the destination page of the click[2]. Then success of the click can be defined using a threshold T as follows:

$$success(c) = \begin{cases} 1, if\ dwell\ time > T\ seconds \\ 0, otherwise \end{cases}$$

Click success is a basic building block of query success, session success, and other "success"-related metrics.

Suppose we want to understand the impact of varying the click dwell time threshold on metric quality: will using 15 seconds, 30 seconds, or 60 seconds thresholds result in better metrics?

The typical approach to answering this question would require collecting labeled log data, where a click in the logs is annotated with success or failure either by the user themselves or by a human judge. One can then compare the success labels from humans to those generated by different cut-off thresholds to determine the best threshold (see Kelly et. al. [14] for several examples of research where this kind of evaluation was conducted).

Obtaining such training data is very expensive, sensitive to the specifics of the judgment process used, and prone to introducing biases. Moreover, this approach only evaluates the accuracy of the click success *definition*. It does not tell us how much the metrics based on this definition have improved, which is what one ultimately cares about. In a way this approach can be thought of as an "offline" evaluation based on labels obtained through a special

[2] Note that this can't always be precisely measured. For example if the destination page is a 3rd party web site, then search engine does not have the data to measure dwell time exactly. In these cases approximations such as time between the click and the return of the user to the search engine can be used.

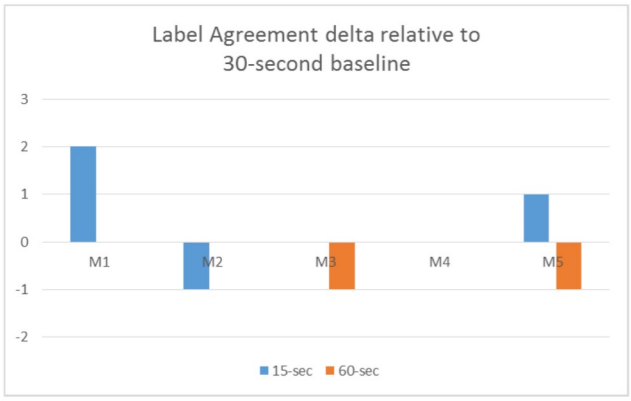

Figure 6. Success threshold evaluation.

labeling process, while the approach we describe below is an "online" evaluation based on real user behavior.

Applying our evaluation approach in this setting is straightforward. We pick several important metrics that use click dwell time definition, and implement modified versions of these metrics using different threshold values: 15, 30, and 60 seconds. We run our evaluation framework and compare the modified metrics to the baseline (threshold of 30-seconds). With "interesting" experiment corpus and Metrics Lab the results can be obtained within 1-2 days.

Figure 6 shows the results of the evaluation based on 5 metrics. Due to the proprietary nature of the success definition used in Bing we do not disclose the details of these metrics and abbreviate them as M1-M5. All of these metrics, however, use the dwell time threshold discussed above. Figure 6 depicts the delta in LabelAgreement, or the gain in the number of experiments the metric gets right if we were to switch to using the new threshold value. No bar (e.g., M4) means the delta is 0.

The 60-second definition is worse or equal to the baseline for all metrics. The 15-second definition is mixed, doing better on two metrics, worse on one metric, and the same on the remaining two metrics. While the 15-second version may seem slightly better, it isn't better across the board and the absolute difference is very small considering that the evaluation was conducted on a 100+ experiment corpus.

While the all-up changes in LabelAgreement are small, one may wonder if there were more individual experiments whose agreement is affected, which just happen to add up to a small overall change. While possible in theory it is not the case here. The changes shown on Figure 6 are the only changes observed in this study. LabelAgreement is identical for all other experiments in the corpus.

Based on this analysis we can conclude that the impact of threshold changes (within reasonable range) on metric quality is small and there is no reason to change the standard 30-second threshold. We conducted similar studies for other thresholds (e.g. 30-minute inactivity threshold for session boundary detection) and found that, within a reasonable range, thresholds rarely have strong impact on metric sensitivity and agreement.

5.3 Measuring User Effort

One of the commonly used ways to measure user effort in satisfying their information need is *Time to Click* [24]. This metric measures the time from the start of a user session (first query), to the first result click. The better the results are and the more clear the page is, the sooner the user will be able to decide where to click.

Table 3. Improvement from switching to long clicks for measuring user effort.

Metric	Percent Improvement
Sensitivity	84%
Label Agreement	200%
Label Disagreement	0%

In this case study we compare *Time to Click* to another, similar metric: *Time to Long Click*, where we define "long" as user not returning to the search engine for at least 30 seconds after the click. Intuitively, a long click should be a better indicator of user actually finding what they wanted. However, some sessions with clicks may not contain any long clicks, and, as discussed in Section 1, a long click is not a perfect success criteria. It is not apriori clear whether counting only long clicks in this metric will improve it, and by how much.

The results are shown in Table 3. Switching to long clicks has dramatic impact on metric quality. Sensitivity almost doubled, Label Agreement tripled, and Label Disagreement remained the same.

Deeper analysis showed that *Time to Long Click* wins in pretty much all feature areas (e.g. quality of web results, user interface improvements, ads, etc.) We observed several cases where the two metric disagreed. Both were statistically significant, but with deltas in different directions. In both cases *Time to Long Click* was correct and *Time to Click* was wrong. Surely, the "small changes can have big impact" rule of experimentation [19] applies to experiment metric development as well.

6. IMPACT

The measurement framework described in this paper has been deployed in Bing for the last several years. It is used by over a dozen Data Scientists, Developers, and Program Managers on a monthly basis. In addition to case studies described above, some examples of the problems it has been applied to are

- Developing ship guidelines for experiments
- Improving "success" metrics
- Evaluating heuristics for more accurate dwell time computation
- Developing revenue/relevance tradeoff metrics
- Developing metrics for "good abandonment"
- Evaluating impact of transformations (e.g. taking a log of a metric value) on metrics

By applying a data driven approach to metric evaluation we have been able to substantially improve our metrics and, as a result, ship better features and improve search experience for Bing's users.

7. CONCLUSION

Good metrics are extremely important for an organization, yet designing good metrics is a difficult process. This paper describes a metric evaluation framework used in Bing to help design good metrics and improve them over time. We define important characteristics of good metrics and propose meta-metrics to evaluate metric sensitivity and alignment with user value. We

discuss the system architecture including performance challenges that need to be addressed in order to enable implementing such a metric evaluation framework in practice. We examine properties of common search metrics, showing how to design better, more sensitive metrics. Finally, we give several examples from our experience of using this metric evaluation framework at Bing to evaluate metric improvement ideas. These examples show that our framework is superior in many respects to the traditional evaluation approaches based on manually labeled data. It is also simple, allowing even non-technical people in the organization evaluate their metric ideas.

Even though in this paper we applied the metric evaluation framework only in the area of web search, it can be applied in any domain, provided an experiment corpus could be constructed.

8. REFERENCES

[1] **Angrist, J. D. and Pischke, J-S.** *Mastering Metrics: The Path from Cause to Effect.* 2014.

[2] **Blackburn, C. and Valerdi, R**. *Navigating the Metrics Landscape: An Introductory Literature Guide to Metric Selection, Implementation, & Decision Making*. Conference on Systems Engineering Research, 2009.

[3] **Davis, J.** *Measuring Marketing: 103 Key Metrics Every Marketer Needs.* 2006.

[4] **Deng, A., Shi, X.** *Data-Driven Metric Development for Online Controlled Experiment: Seven Lessons Learned*. Conference on Knowledge Discovery and Data Mining, 2016.

[5] **Deng, A., Xu, Y., Kohavi, R. and Walker, T.** *Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data*. Conference on Web Search and Data Mining, 2013.

[6] **Farris, P. W., Bendle, N. T., Pfeifer, P. E. and Reibstein, D. J.** *Marketing Metrics: The Definitive Guide to Measuring Marketing Performance.* 2010.

[7] **Hassan, A., Jones, R. and Klinkner, K.L.** *Beyond DCG: user behavior as a predictor of a successful search.* Conference on Web Search and Data Mining, 2010.

[8] **Hassan, A., Shi, X., Craswell, N. and Ramsey, B.** *Beyond Clicks: Query Reformulation as a Predictor of Search Satisfaction.* Conference on Information and Knowledge Management, 2013.

[9] **Hauser, J. and Katz, G.** *Metrics: you are what you measure!* European Management Journal, 1998.

[10] **Hohnhold, H., O'Brien, D., Tang, D.** *Focus on the Long-Term: It's better for Users and Business.* Conference on Knowledge Discovery and Data Mining, 2015.

[11] **Jarvelin, K. and Kekalainen, J.** *Cumulated gain-based evaluation of IR techniques.* ACM Transactions on Information Systems 20(4), 422–446, 2002.

[12] **Jiang, J., Hassan, A., Shi, X. and White, R.** *Understanding and Predicting Graded Search Satisfaction.* Conference on Web Search and Data Mining, 2015.

[13] **Kaplan, R. and Norton, D.** *The Balanced Scorecard - Measures that Drive Performance*. Harvard Business Review, Vol. 70, Issue 1, pp. 71-80, 1992.

[14] **Kelly, D. and Teevan, J.** *Implicit feedback for inferring user preference: A bibliography.* ACM SIGIR Forum, 37(2), pp. 18-28, 2003.

[15] **Kerr, S.** *On the folly of rewarding A, while hoping for B*. Academy of Management Executive, Vol. 9, No. 1, pp. 7 - 14, 1995.

[16] **Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R.** *Controlled Experiments on the web: survey and practical guide.* Data Mining and Knowledge Discovery journal, Vol 18(1), pp. 140-181, 2009.

[17] **Kohavi, R., Deng, A., Frasca, B., Longbotham, R., Walker, T. and Xu, Y.** *Trustworthy Online Controlled Experiments: Five Puzzling Outcomes Explained*. Conference on Knowledge Discovery and Data Mining, 2012.

[18] **Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y. and Pohlmann, N.** *Online Controlled Experiments at Large Scale*. Conference on Knowledge Discovery and Data Mining, 2013.

[19] **Kohavi, R., Deng, A., Longbotham, R. and Xu, Y.** *Seven Rules of Thumb for Web Site Experimenters*. Conference on Knowledge Discovery and Data Mining, 2014.

[20] **Li, J., Huffman, B. and Tokuda, A.** *Good Abandonment in Mobile and PC Internet Search.* Special Group on Information Retrieval (SIGIR) Conference, 2009.

[21] **Li, P., Burges, C. and Wu, Q.** *Learning to Rank Using Classification and Gradient Boosting*. Conference on Neural Information Processing Systems (NIPS), 2007.

[22] **Marr, B.** *Key Performance Indicators (KPI): The 75 measures every manager needs to know.* 2012.

[23] **Ridgeway, V. F.** *Dysfunctional consequences of performance measurements*. Administrative Science Quarterly, Vol.1, Issue 2, pp. 240–247, 1956.

[24] **Sadeghi, S., Blanco, R., Mika, P., Sanderson, M., Scholer, F., and Vallet, D.** *Predicting Re-Finding Activity and Difficulty*. European Conference on Information Retrieval, 2015.

[25] **Schmenner, R.W., and Vollmann, T. E.** *Performance Measures: Gaps, False Alarms and the "Usual Suspects"*. International Journal of Operations and Production Management, Vol. 14, No. 12, pp. 58-69, 1994.

[26] **Tang, D., Agarwal, A., O'Brien, D. and Meyer, M.** *Overlapping Experiment Infrastructure: More, Better, Faster Experimentation.* Conference on Knowledge Discovery and Data Mining, 2010.

[27] **Yi, X., Hong, L., Zhong, E., Liu, N. and Rajan, S.** *Beyond Clicks: Dwell Time for Personalization*. ACM Conference on Recommender Systems, 2014.

9. ACKNOWLEDGMENT

The authors would like to thank Brian Frasca, Ron Kohavi, Toby Walker, Georg Buscher, and Widad Machmouchi for their help developing the metric evaluation framework and their feedback on the drafts of this paper. Special thanks to Brian Frasca for suggesting the title of the paper.