Delivering Relevant Results For Search Queries With Location Specific Intent

Anand Chakravarty
anandmc@microsoft.com
John Guthrie
johngut@microsoft.com

Abstract

With a still growing number of users and an ever-growing volume of information available, the Internet presents many interesting and continuously morphing challenges in the area of Search. In addition to the scale of data to be processed, the results of user search queries are expected to be highly relevant and delivered speedily. Presenting these results to the user in a manner that enables them to decide and act based on the information provided is a primary characteristic of a good Search engine. Measuring the accuracy and relevance of Search results is thus an important area in the Testing of Search engines. Considering the high volume of information to be processed, the extremely diverse nature of query-intent and the growing expectation of high relevance from Search users, testing of Search engines requires the traditional QA characteristic of passion for quality, combined with a high-level of comfort with ambiguity and strong automation skills.

When we consider the area of search queries that have a location specific intent, there are other factors that become important along with the usual technical problems. Most queries that have local intent have a higher degree of immediacy in terms of the user's intent to act on the results returned to their queries. There is thus greater expectation of the results being fresh and accurate. With the growing market for Mobile devices, searches with local-intent are becoming more popular. As a tester, when presented with measuring how well a Search engine performs for such queries, it is important to understand the scale and variety of queries involved. Because there is a high level of ambiguity and variation in search queries and results, statistical metrics are a natural tool to measure Search engine quality.

In this paper we cover the methods used to obtain metrics for measuring relevance of search queries with local intent. Testing is done while fundamental components are in a dynamic state: rankers, intent detectors, content, location identifiers, etc. A QA team that comes up with good metrics to measure the quality of search results in such scenarios increases the quality of Search Experience delivered to their users, and helps to evolve the quality of solutions implemented in this extremely challenging problem space.

Biography

Anand Chakravarty *is a Software Design Engineer in Test at Microsoft Corporation. He has been testing online services for most of the past decade, and is currently on the Bing Local Search Team.*

As a serial entrepreneur, ***John Guthrie*** *founded, sold, and/or participated in the IPO of companies in various fields. He has led teams to develop, test, and patent technology for Digital Rights Management, Data Storage, and Web Site Modifications. Now at Microsoft, he is working on ensuring the quality of the results for local-intent queries at Bing.*

1. Introduction

User-perceived relevance of web-based Search results is a key metric that defines the success of a Search engine. It is therefore vitally important that the metric be correctly and clearly defined, accurately and systematically measured and, finally, evaluated and improved over time.
To understand what is meant by user-perceived relevance of a Search result, let us consider, as an example, the Search query "Pacific Northwest software quality". The first question to answer here is: what is the user's intent when they search for "Pacific Northwest software quality"? Without a lot of background, it could be assumed that the user might be looking for:

- an organization/corporation by that name,
- or a conference by that name,
- or book with that title,
- or a study explaining why software quality in the pacific northwest is lower/higher,
- or a recent news item relating to software quality in the region, and so on.

Once the Search engine has arrived at some judgment as to user-intent, it then searches its web-index (a database storing the Search engine's snapshot of the Internet) for web-pages that are relevant to that intent. Let us say that, in this case, the user-intent was about a software quality conference or organization in the Pacific Northwest. And let us say the Search engine identified that user-intent correctly. Based off that intent, the Search engine looked up pages in its web-index and then returned the results showed in Figure 1:
The next question to answer is: how relevant are the results that are returned for the query. Let us consider the results shown in Figure 1. The first returned result for this query, which is a link to www.pnsqc.org, is highly relevant. The 2nd and 3rd results, which are links to previous conferences, are also relevant, although less so than result number 1. The 4th returned result, which is a link to an Art Design web-site, is irrelevant to the assumed user-intent. While the returned page does contain the terms Pacific Northwest and Quality, it is not related to software. The 5th result is also relevant, although it is another organization making a reference to an older PNSQC conference.
From a user's perspective then, result number 1 is the most relevant. Results number 2 and 3 could be improved by replacing them with more recent conferences. Result number 4 should have been ranked lower. Result number 5 should have replaced result number 4, and other more recent conferences should probably have been returned and ranked higher than result number 5 as well.
In terms of rating the results then, we could rate result number 1 as Perfect, results number 2 and 3 as Good, result number 4 as Bad and result number 5 as Good.
There are thus two features in measuring the relevance of results returned for a particular query: the degree to which a result matches the user's intent, and the position (rank) of a result in the set of returned results. A highly relevant result returned at a lower position is clearly less useful to the user, same as a less relevant result being returned with a high rank.
The parameters impacting the user-perceived relevance metric that have been defined so far are fairly static. The dynamic nature of the Web, and of the reality captured on the Web, add another level of complexity to accurately defining and measuring user-perceived relevance. Let us consider another example to understand that, in this case the query "U.S. Open results". Depending on the time of the year this query is performed, the user-intent might be to see results for the U.S. Open Golf Championship, or the U.S. Open Tennis tournament. A Search engine that returns different results for that query depending on which event is closer to the time the query is performed would be more returning more relevant results to a user

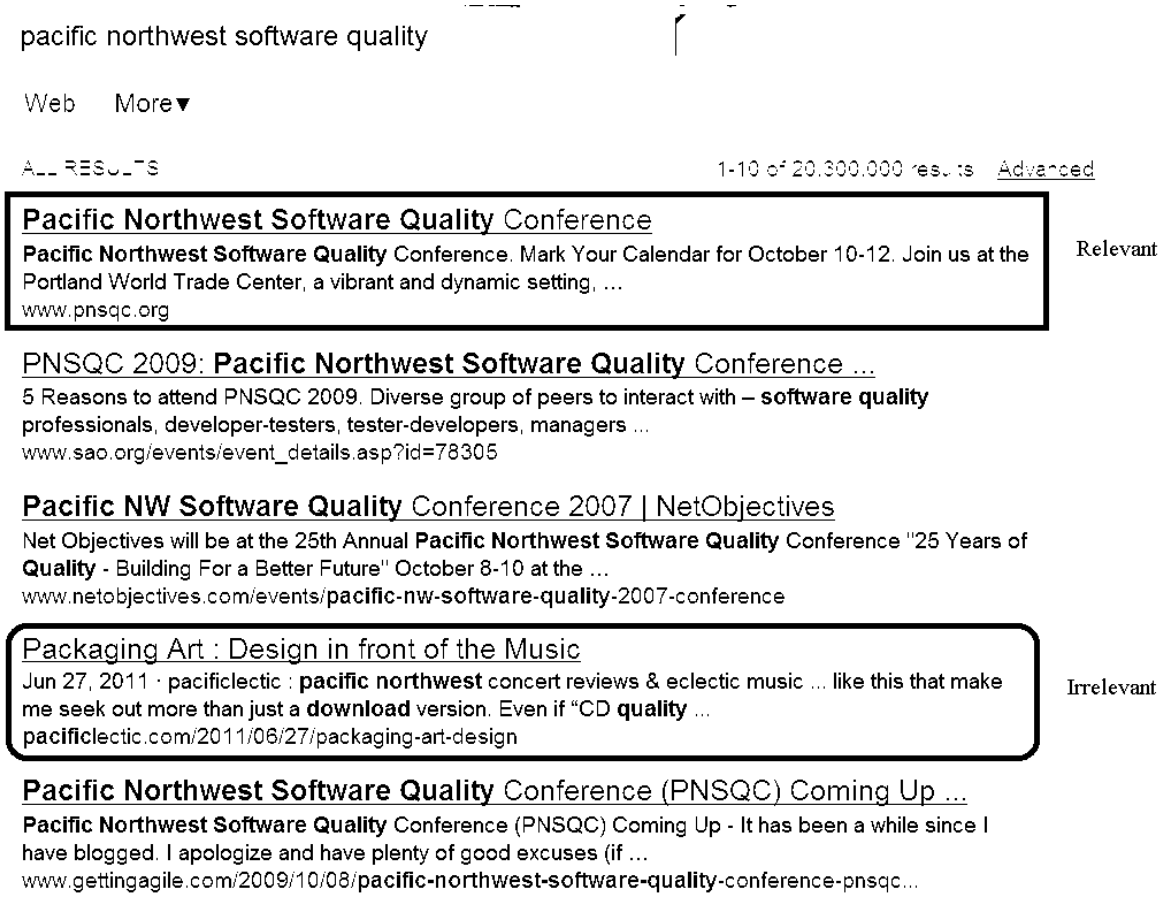


Figure 1

Quantifying the metric of user-perceived relevance involves rating of results for Search queries. Clearly, the number of unique Search queries that a Search Engine would receive is a very large population and it is computationally infeasible to enumerate all queries. For testing and measuring relevance, we therefore use a sample of all queries occurring in the real-world. A well-defined sample would closely match the distribution of queries in the real-world and also account for the frequency with which a query occurs. This means that it would have a mix of popular and rarely used queries, and there would be a weight associated with each of them, the more popular queries getting a higher weight. There are implications of working with only a sub-set of actual queries, because a sub-set is never going to comprehensively match the real world. These shortcomings should be correctly accounted for in calculating relevance metrics. With that in place, the sub-set of queries are then run through the Search engine and the results obtained for those queries should then be rated by human judges. These ratings could involve assigning a number to each returned result, such as 2 for Perfect, 1 for Good, and 0 for Bad. The guidelines for rating should be clear, logical and easily understood. . Also, because the top results returned for a query are more important than results returned at lower positions, only a limited number, say 5 or 10, of the top results for a query should be considered in measuring user-perceived relevance.

Once we have the metric thus quantified, they then become a key input for driving improvements in the Search Engine. To effectively drive such improvements, the collection, storing and evaluation of these metrics should be done in a well-organized and repeatable manner. The variables affecting the calculation of the metrics should be minimized. The numbers reported should be reusable over multiple iterations and across multiple sets of queries. And the trends of these metrics over time and different versions of the Search Engine should be easily viewable to help understand the impact of different changes and to evaluate the progress made over time.

All of the above are applicable to a wide-range of Search Queries. There is a certain category of queries that however require further tuning of the way their user-perceived relevance is calculated. These are queries that have a Local intent. Consider 2 simple queries: ".NET garbage collection" and "pizza". The relevance of results returned for the first query is independent of the location from which a user performs the query. In contrast, for the second query, the relevance of results returned for it vary according to the location of the user. For a user from Portland searching for "pizza", results located in Seattle would be irrelevant, and vice-versa.

In this paper, we explain in detail some of the methods used by the Bing Local Search team to measure user-perceived relevance of Bing search results for Local intent queries. We start by covering the creation of query-sets to measure user-perceived relevance. Once the query-sets have been created, they are run through the Search Engine and results are obtained and stored. These results are then evaluated by judges. The judges follow a set of guidelines that helps them rate results returned for a query. The guidelines capture the steps that we used above to identify relevant and irrelevant results for the query "Pacific Northwest software quality". Each result gets a score, which could be a number like 1 through 5 or a tag, say Perfect, Good, etc. A combined rating for the results returned for a query and the position at which each is returned are then a measurement of user-perceived relevance for that set of queries for the version of Search Engine that was tested. The approach and methodologies we have used have helped accurately understand the relevance of results returned to users performing queries with Local intent over multiple versions of the Search Engine.

2. Creating Query Sets for Measuring Relevance

The number of unique Search queries performed by Internet users is of an extremely high order, even if the domain is restricted to queries with Local Intent. Analysis and evaluation of Search engines must then, by necessity, work on subsets of the entire corpus of possible queries. For testing the first version of a Search engine, in the absence of real world data, a realistic estimate of expected queries is essential. For subsequent versions, data of queries from real-world usage of the Search engine helps make more accurate evaluation of Search engine relevance. A mechanism to log such queries for this purpose, while respecting user-privacy and maintaining anonymity, is therefore a requirement. Privacy of users is maintained by not storing the source IP-address of the query, and only storing the query text and the query location at city-state level.

From the available set of real-world queries, we then create a set of sampled queries to be used for measuring user-perceived relevance. A random sampling ensures a better reflection of real world distribution and thus will eventually yield more accurate metrics.

In our analyses of Search queries, it was found that distribution of query-frequency fits a Zipfian distribution. At the head of the curve are popular queries that contribute to a high percentage of Search traffic. They are followed by a long tail of unique queries that are searched for with low frequency. A sample distribution is shown in Figure 2:

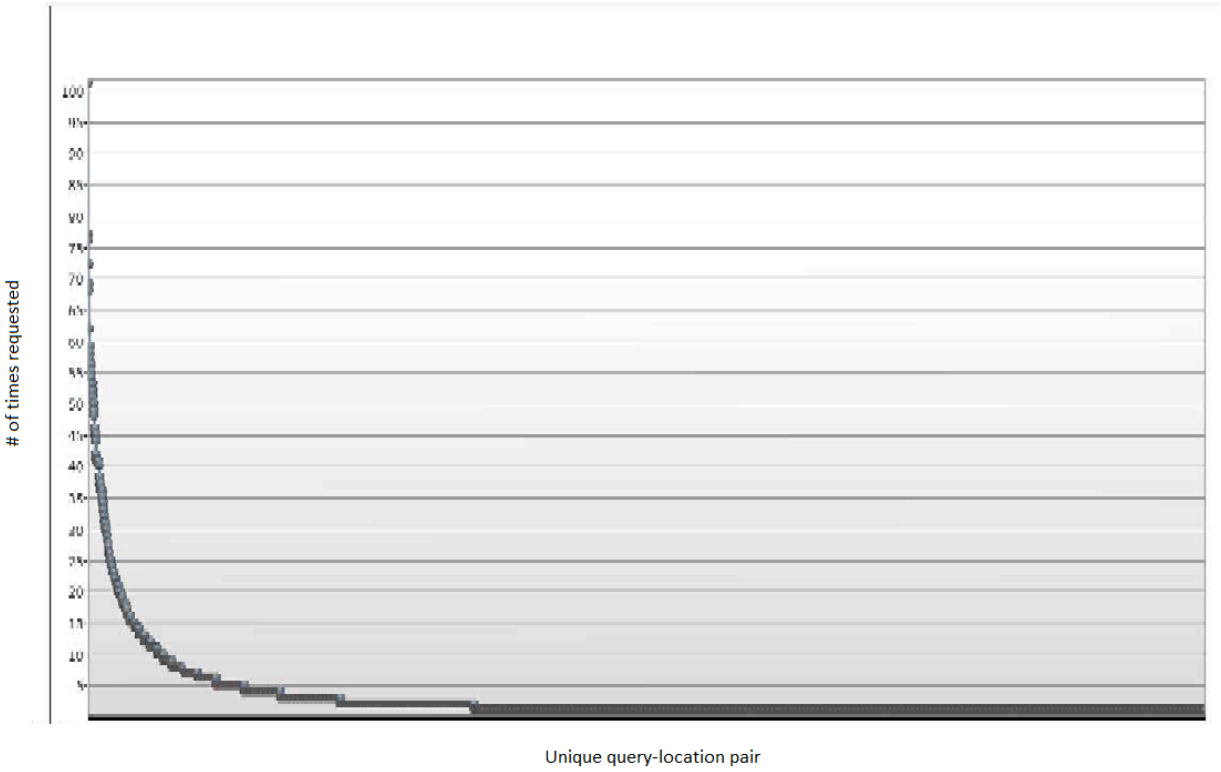


Figure 2

To properly account for this distribution, we apply weights to the sampled queries depending on the frequency with which a particular query occurs. Queries that have a higher frequency thus contribute more to the metrics measuring user-perceived relevance. This approach helps the Search engine be better tuned for higher-frequency queries while continuing to measure relevance for queries that occur with much less frequency.

Queries with local-intent require some additional considerations in creating query-sets in order to correctly to measure the relevance of results returned for such queries. The first consideration is measuring the correctness of the Search Engine in identifying queries that have local intent. One extreme design is for a Search Engine to assume that all queries it receives have local-intent and then try to obtain results for all queries. This obviously is problematic both in terms of returning relevant results for queries that do not really have local-intent, and in terms of the performance implications of using components of the Search Engine that do not have to be exercised. Based on analysis of query-patterns and observed usage, we could arrive at an expected percentage of total traffic that realistically has local-intent. Query-sets created for measuring relevance of local-intent queries should have a mix of queries with local and non-local intent that match the real-world distribution of local and non-local intent queries. One of the metrics calculated should include the percentage of queries from the set of input queries that are triggered as having local-intent.

Another consideration in creating query-sets for measuring relevance of local-intent queries is the actual location of the query. Let us consider an example of such a query, “medical clinics”. Depending on the location from where this query is issued, the user expects different results. An example of a query with non-local intent is “u.s. open winner”. For the latter query, it is safe to ignore the location, because the expected results are independent of the geographical source of the query/user. In creating a query-set to measure relevance of queries that do not have local intent, “u.s. open winner” with a location of “Chicago, IL” is the same as “u.s. open winner” with a location of “Los Angeles, CA”. For measuring relevance of queries that have local-intent however, “medical clinics” from “Chicago, IL” is different from “medical clinics” from “Los Angeles, CA”. Therefore, in a query-set for measuring relevance of queries with non-

local intent, “u.s. open winner” would only occur once, whereas in a query-set measuring relevance of queries that do have local intent, the query “medical clinics” could appear more than once, each time with a different location. If the query-set were to contain “medical clinics” only once, with just one unique location, then the metrics won’t capture the reality of users performing the query from different locations. There is therefore a likelihood that the metrics will either miss improvements made to the Search Engine for such queries, or incorrectly report improvements for such queries based on just one location. This is an important distinction in the methodology of creating query-sets to measure relevance of queries with local intent.

There is another variation in local-intent queries that must be taken into account for creating query sets to measure their relevance. That is the difference between implicit and explicit query-location. In the query “medical clinics”, the location for which results are expected is implicit. By contrast, the query “hotels in Las Vegas” has the location of expected results specified explicitly in the query. So while the latter query does have local intent, the results expected are independent of the location of the user making the query. i.e., a user in Florida would expect the same results for “hotels in Las Vegas” as a user in Iowa. Query-sets for measuring relevance of local-intent queries should have a realistic distribution of queries with both implicit and explicit locations.

In addition to the above broad considerations in creating query-sets for measuring relevance of local-intent queries, there are other features that are useful to take into account depending on the focus and priorities of the Search Engine. A search engine might have different versions depending on the market for which it is designed. A different query set would be required to measure relevance in each of those markets. At the same time, the same search engine might be used to measure relevance of local-intent queries independent of location. For example, the same version of a search engine could be used to answer a query for a user in, say, Boston MA, for local-intent queries with relevant results located in the US, such as {Italian restaurants in Los Angeles} and for queries with relevant results located outside the US, such as {hotels in Melbourne Australia}. For such search engines, the query-set should have a realistic mix of queries across all supported geographical locations.

For a search engine that specializes in queries relating to a particular category of business, such as local queries dealing with a strong restaurant-intent or a strong hotel-intent, it is essential to test with query sets that are predominantly made up of queries matching those intents. This involves identifying not just the local-intent for a query, but also the specific business intent. Classifiers that identify such targeted intents are useful in identifying these queries and help in creating query-sets for measuring relevance of queries matching the targeted intent.

3. Measuring Gain, Precision and Recall

Using the query-sets created according to the considerations outlined above, we are now ready to measure the relevance of a Search Engine for that set of queries. To quantify user-perceived relevance, we calculate numbers for 3 metrics: the discounted cumulative gain (DCG) in relevance for a set of queries, the Precision of results returned for a query and the Recall of results relevant to a query.

Consider a query q, for which a search engine returns 5 results, r_1 through r_5. The cumulative gain of the relevance of returned results for a certain position for this query is the sum of the graded relevance of each result up to this position for the query. To simplify measurements, let us assume a result is graded only as either relevant or irrelevant. i.e., relevance values are binary. For such a query, if only 3 of the 5 returned results are relevant, its cumulative gain at position 5 is then 3.

The discounted cumulative gain (DCG) metric helps to account for the position of relevant results for a query. From a user’s standpoint, search results are more useful if a more relevant result is shown higher than a less relevant result. The DCG metric includes the graded relevance of a result and the position at which it occurs.

Because of the variety and distribution of queries received by a Search Engine, the DCG metric should be normalized so that it may be compared across a wide-range of queries. The normalized DCG, or NDCG, for a set of queries at a particular position is obtained by dividing the DCG of results returned up to that position by the DCG of the ideal set of results at that position for the query. The ideal DCG is obtained by

sorting the results for a query by relevance, with the most relevant result at the top and going down to less relevant results. For perfectly ranked set of results, the normalized DCG is therefore equal to 1.
Precision indicates the relevance of a result; the more relevant a result, the higher its Precision value.
Recall measures the completeness of the results returned for a Search query. A result set for a query is said to be complete if all relevant results for that query are returned in that set. In our example query "Pacific Northwest software quality", a complete set of relevant results could be all web-pages relating to the PNSQC. Let us say there are fifty such pages. If a Search Engine returns only 10 of those pages in its set of results for the above query, then its recall for the query would be calculated as 10/50 = 0.2. A higher recall indicates that more of the relevant results for a Search query were returned.

Given the nature of Search Engines, there is a tendency for Precision and Recall to push against each other such that higher recall comes at the expense of lower precision and vice versa. This happens because as the Search Engine attempts to uncover more results, it also surfaces results whose relevance is questionable. Conversely, if the Search Engine is stricter about what it considers a relevant result, it may miss some results on the fringe. As with most statistical quantities, it is important to have accurate margins of error calculated for each of these metrics, calculated for the system being tested.

Using the well-designed query sets defined in the previous section, calculating the gain, Precision and Recall values for the result obtained for those queries by a Search engine yields a good measurement of the relevance of results for a Search query.

It should be emphasized that like most metrics, the numbers that arise out of their calculations are essentially just an indication of overall performance, they do not tell us a lot until we delve into the details. Let us consider a scenario where for a particular version of the Search engine, when it was tested with a set of queries, there was a drop in the overall Recall numbers. It is the nature of a non-deterministic system that for the same input, it won't always produce the same output. Also, for a different version of such a system, the difference in results for the same input will also be different, in a non-deterministic way, for the individual items in the input. Therefore, in the scenario under consideration, not all queries would experience the same drop in Recall: some queries might even see an increase in Recall with a new version, while most may see a drop. To identify the cause of this drop in Recall, we would first identify queries that experienced a drop in their Recall. Then we look at each of these queries and try to identify the root-cause of the drop. As we analyze more queries, we should place queries into buckets, with each bucket mapping to a particular root-cause. Once we are done with identifying the root-cause for all queries that experienced a drop in Recall, we then look at the number of queries impacted by a particular root-cause. This then drives improvements of the Search engine, as we fix the root-causes depending on the scope of their impact.

In a smoothly running system with a high degree of repeatability, the numbers are expected to be more reliable. In products developed under the Agile model where a lot of components change simultaneously, these metrics tend to show a higher level of variability, and it is then more important to look at trends and apply accurately calculated margins of error to these metrics.

Furthermore, as the Search Engine goes through multiple iterations, it becomes essential to recalculate new baseline values for these metrics, both to measure the impact of each new version and to better understand trends in Search Engine quality over long time-spans. Additionally, as usage patterns change, the set of input query-sets may also require modifications. Such updates should also drive recalculating of baseline values for the metrics.

4. Guidelines for judges

The process of measuring user-perceived relevance ultimately involves human evaluation of results for Search queries. These are performed by a team of human judges. The judges are shown either a query with all the results returned for it or a query-result pair. They rate either the whole set of results or each query-result pair. In order for the ratings to be accurate, a well-defined set of guidelines for rating each result should be provided to the judges. These guidelines would be different for each domain of Search. For example, for the component of a Search engine that is focused on providing results for queries relating to News articles, the guidelines might include a definition of freshness of a result. Similarly, for queries related to Movies, the guidelines could include defining the completeness of information

presented in the results: do the results contain show-times, and ratings? For queries with Local intent, as mentioned earlier, it is important to consider the location for which a query is performed. A high-level overview of guidelines used for rating results for Local Search queries is shown in Table 1. The table shows example queries and a result for each. Using a scale of Perfect, Excellent, Good, Fair and Bad, the table explains to the judges what rating would be appropriate for each example. The judges then use these guidelines to manually rate all queries and corresponding results that are presented to them as part of the relevance measurement process. These ratings are then used in calculating the SNDCG, Precision and Recall metrics for each query of a set of queries. As the Search engine evolves, and data in the real world changes, these guidelines may have to be updated and fine-tuned so that the ratings given by the judges remain accurate and useful for the purposes of calculating relevance metrics.

Query	Local Search result returned	Rating
{Ikea in Seattle, WA}	Ikea at 601 SW 41st Street, Renton, WA	"Perfect"; The nearest location of a specialty store
{dilbeck realty los angeles CA}	Dilbeck Realtors GMAC Rl Est 2486 Huntington Dr, San Marino, CA 91108	"Excellent"; A relevant result close to Los Angeles was returned.
{lane bryant} from Elmhurst,Illinois	Lane Bryant 101 Oakbrook Ctr, Oak Brook, IL 60523	"Good"; Result matched query, although it was not exactly at the specified location.
{ Pets Mart Seattle}	Pets Mart at 1203 N Landing Way, Renton, WA	"Fair"; The Pets Mart franchise has a location in the city of Seattle
{Hong Kong Express Seattle}	Hong Kong Airlines	"Bad"; The result is not relevant to the query. The user is looking for a restaurant and is returned an airline.

Table 1

It is important to be aware of the subjective nature of these ratings. The same query-result pair could get a different rating depending on the human judge who is assigning the ratings. To account for this, we work with a large set of queries, and assign multiple judges to a query-result pair, which helps to smooth out such variations between judges. Such variations also reflect real-world user-perception of relevance of Search results, and are therefore acceptable to a certain extent.

5. Obtaining and analyzing the metrics

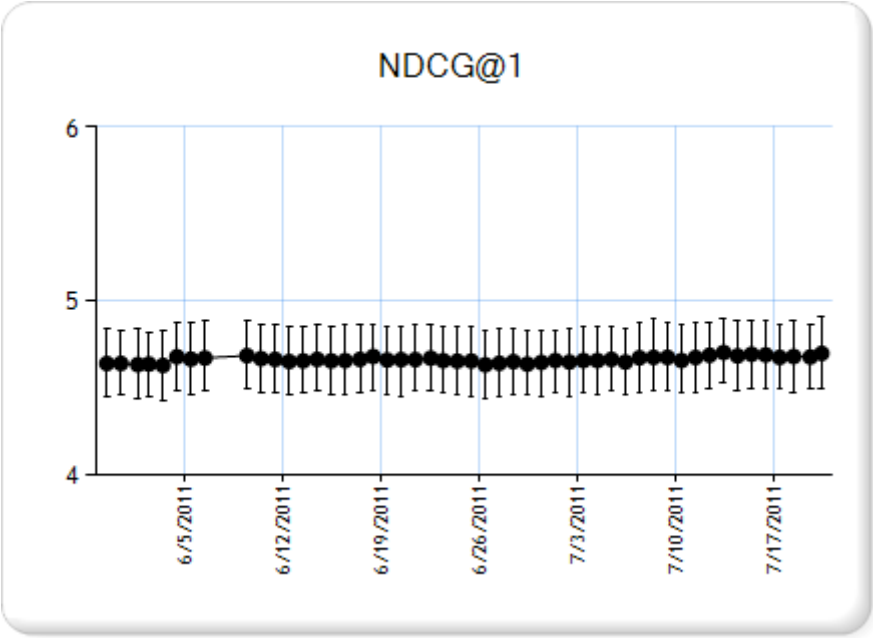
At this point, we have obtained the queries that represent real-world usage, defined the metrics used to measure user-perceived relevance and provided guidelines to human judges to rate results for Local Search queries. The next step is to actually use the queries to obtain results, get them judged and then calculate the metrics. The metrics are calculated for multiple versions of the Search engine.

To obtain results for a set of queries, we create automation tools that run each query through the specified Search Engine version and obtain results for each of them. A database to store results for each execution created, which is useful in evaluating trends over time and also for any subsequent investigations and analyses.

The results are then presented in a visual manner to a group of judges, who have also been trained in the guidelines to use for rating results for a query. The rating for each query-result pair, or a query-results set, is stored in a database. Using the ratings, the NDCG, Precision and Recall for each query is calculated. These metrics are then calculated for the complete set of queries, assigning weights to each query depending on its frequency; queries that are more popular get a higher weight assigned when calculating their metrics. This helps us to account for using only a sub-set of all queries encountered in the real-world.

The process of obtaining results for a set of queries, getting them judged and then calculating the metrics should be automated for easy repeatability. Once that process is working correctly, we may run it

regularly over different versions of the Search engine and see how the metrics are trending over time. We may also run the process against 2 separate versions of the Search engine and compare the metrics for the 2 to evaluate differences and improvements over time. Figure 3 shows the changes in DCG, Precision and Recall over time for a fixed set of queries hitting a Search engine.



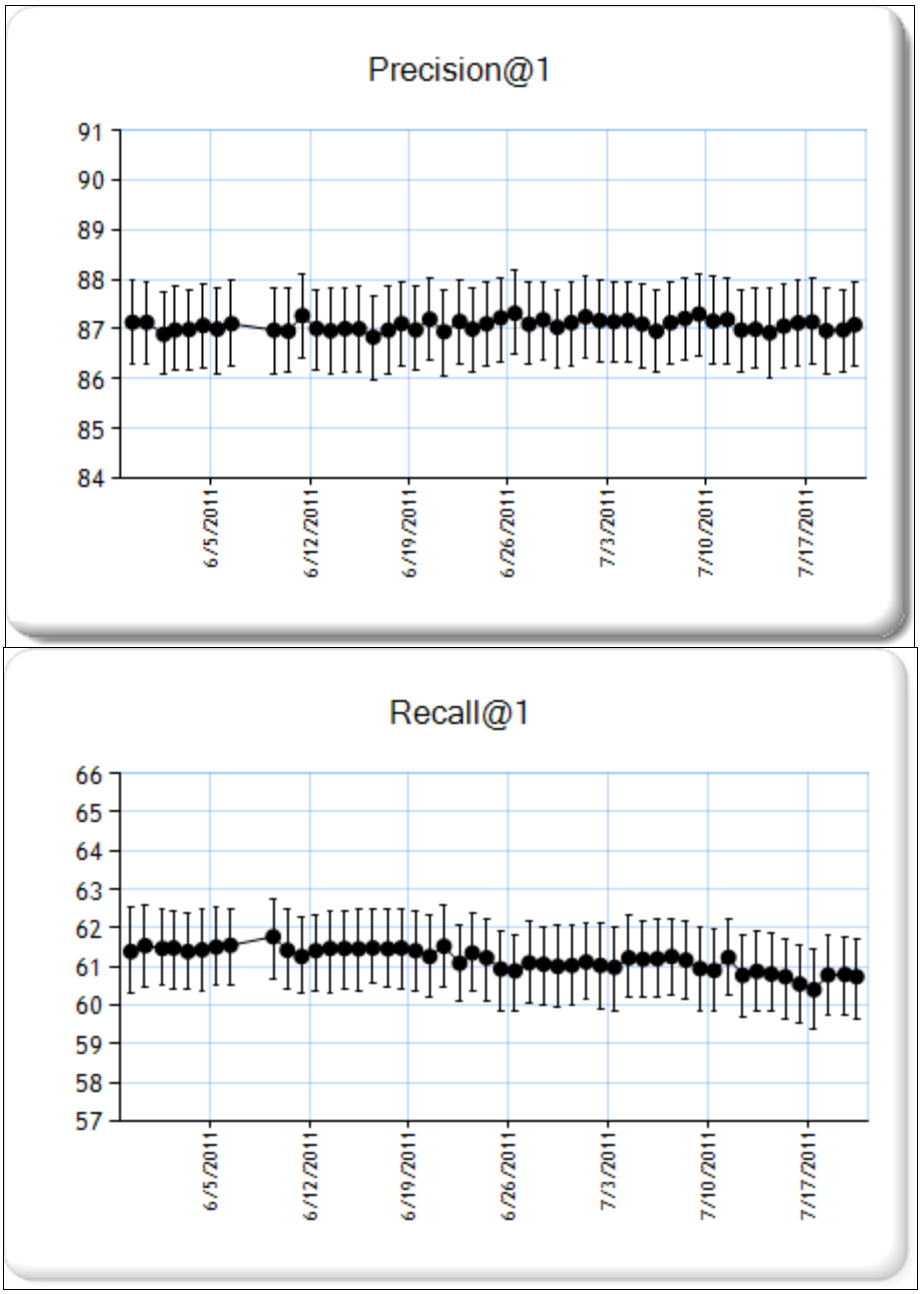


Figure 3

6. Challenges in calculating relevance metrics

As with most Machine Learning systems, there are challenges to quantifying the results of testing a Search engine for location-specific queries. The results returned for the same query changes over time. The calculation of metrics should account for that change. It should also capture the fact that there hasn't been any change when a change is expected. For example, if a restaurant closes, then a Search engine that used to return it among the results for a query looking for restaurants in that area should stop returning it. In terms of relevance metrics, such a change should not affect the Recall value for that query, and if the restaurant continues to be returned, then its Precision value should drop.

Depending on the design, there may also be multiple components that affect the results of location-based Search queries. These could include a spell-checker, a query location identifier, index servers, the data sources and publishing pipeline, etc. Let us consider the query {pizza hug}, in which there is very likely a typo. A spell-checker would usually catch that and return results for {pizza hut}. However, a new version of the spell-checker fails to catch it, or catches it but fails to identify it owing to a misconfiguration or performance issue, then it would impact the relevance metrics calculated for that query. Similarly, changes to the location-identifier might cause Precision and Recall numbers to drop if those changes result in incorrect locations being returned.
All of these moving parts impact the accuracy of relevance metrics. Accounting for them involves dealing with a high-level of ambiguity in the overall system being tested, and correcting the numbers reported so that we have a realistic measurement of user-perceived relevance.

References

Wikipedia, *Precision and recall,* http://en.wikipedia.org/wiki/Precision_and_recall