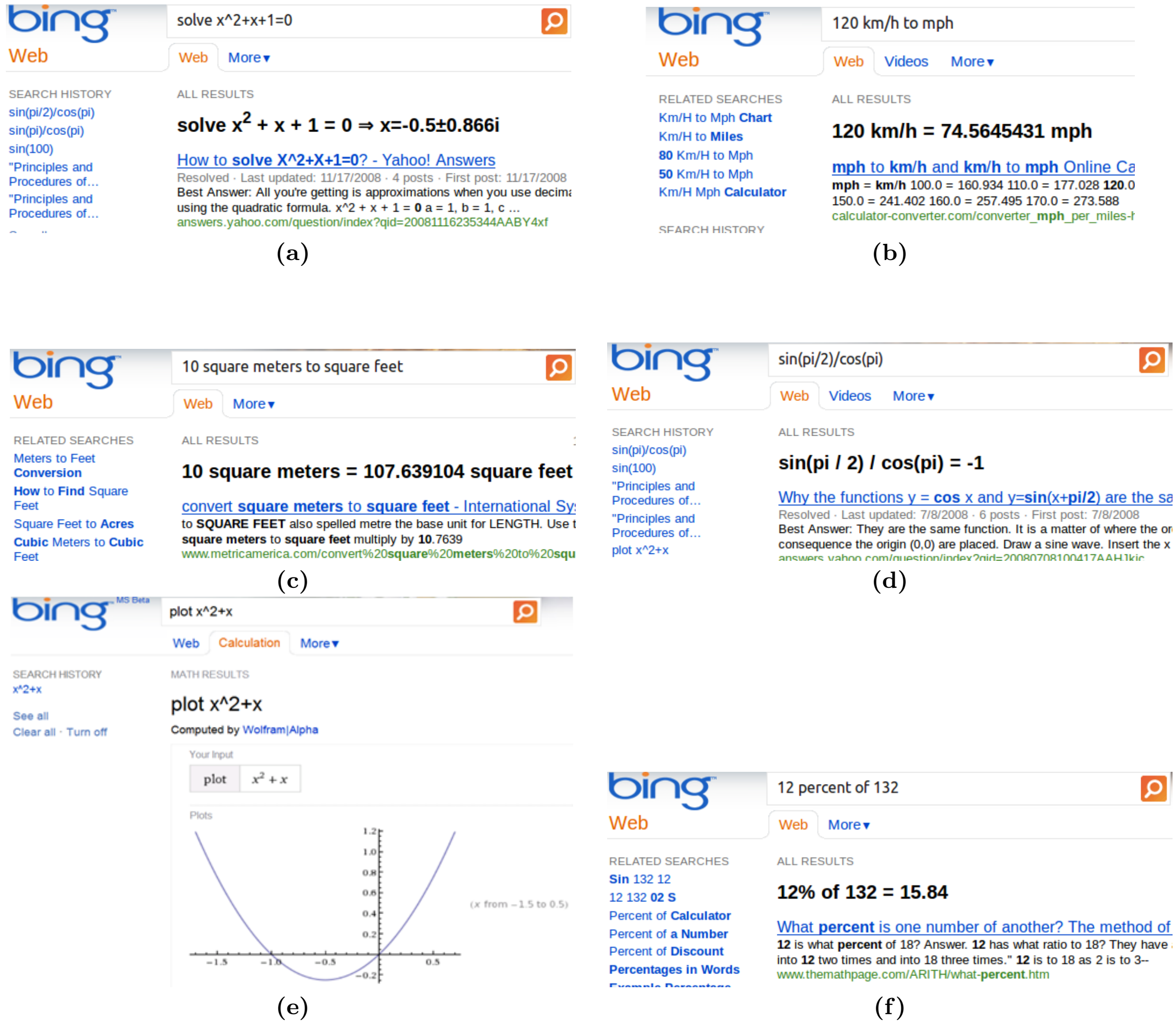


(a)

solve x^2+x+1=0

sin(pi/2)/cos(pi)
sin(pi)/cos(pi)
sin(100)
"Principles and Procedures of…
"Principles and Procedures of…

solve $x^2 + x + 1 = 0 \Rightarrow x=-0.5\pm0.866i$

How to **solve X^2+X+1=0**? - Yahoo! Answers
Resolved · Last updated: 11/17/2008 · 4 posts · First post: 11/17/2008
Best Answer: All you're getting is approximations when you use decima
using the quadratic formula. x^2 + x + 1 = **0** a = 1, b = 1, c ...
answers.yahoo.com/question/index?qid=20081116235344AABY4xf

(b)

120 km/h to mph

Km/H to Mph **Chart**
Km/H to **Miles**
80 Km/H to Mph
50 Km/H to Mph
Km/H Mph **Calculator**

120 km/h = 74.5645431 mph

mph to **km/h** and **km/h** to **mph** Online Ca
mph = **km/h** 100.0 = 160.934 110.0 = 177.028 **120**.0
150.0 = 241.402 160.0 = 257.495 170.0 = 273.588
calculator-converter.com/converter_**mph**_per_miles-

(c)

10 square meters to square feet

Meters to Feet **Conversion**
How to **Find** Square Feet
Square Feet to **Acres**
Cubic Meters to **Cubic** Feet

10 square meters = 107.639104 square feet

convert **square meters** to **square feet** - International Sy
to **SQUARE FEET** also spelled metre the base unit for LENGTH. Use t
square meters to **square feet** multiply by **10**.7639
www.metricamerica.com/convert%20**square**%20**meters**%20to%20squ

(d)

sin(pi/2)/cos(pi)

sin(pi)/cos(pi)
sin(100)
"Principles and Procedures of…
"Principles and Procedures of…
plot x^2+x

sin(pi / 2) / cos(pi) = -1

Why the functions y = **cos** x and y=**sin**(x+**pi/2**) are the sa
Resolved · Last updated: 7/8/2008 · 6 posts · First post: 7/8/2008
Best Answer: They are the same function. It is a matter of where the or
consequence the origin (0,0) are placed. Draw a sine wave. Insert the x
answers.yahoo.com/question/index?qid=20080708100417AAHJkic

(e)

plot x^2+x

x^2+x

plot x^2+x

plot $x^2 + x$

(f)

12 percent of 132

Sin 132 12
12 132 **02 S**
Percent of **Calculator**
Percent of **a Number**
Percent of **Discount**
Percentages in Words
Example Percentage

12% of 132 = 15.84

What **percent** is one number of another? The method of
12 is what **percent** of 18? Answer. **12** has what ratio to 18? They have
into **12** two times and into 18 three times." **12** is to 18 as 2 is to 3--
www.themathpage.com/ARITH/what-**percent**.htm

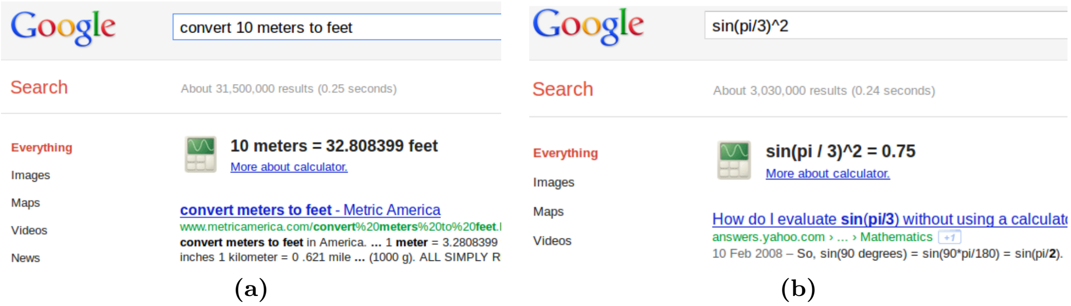
Figure 1: Snapshots of Bing's result page for (a) "solve xˆ2+x+1=0", (b) "120 km/h to mph", (c) "10 square meters to square feet", (d) "sin(pi/2)/cos(pi)", (e) "plot xˆ2+x", and (f) "12 percent of 132".

Although we intuitively defined math queries as a class of queries that should be calculated to produce an answer, a formal definition that allows automatic identification of such queries is still missing. A formal definition should consist of a set of grammars that describe the entire range of math queries. Such a comprehensive set of grammars is unavailable, but it is necessary to identify math queries and distinguish them from non-math queries that are submitted to an engine. It is required to indicate the fraction of math queries that an engine can process and answer as a key evaluation metric, and also to determinine classes of math queries that are not answered, in order to develop modules for processing them and enhancing the engine.

The lack of a comprehensive set of grammars implies that the structure of some math queries are unknown to us. Moreover, math queries typically consist of very few words, and many math symbols, letters, and numbers instead. Therefore, many queries exist that are not math even though they look like math, hence they are difficult to distinguish from math queries. For example model and part numbers (e.g. "lg 120" and "123a-5"), phone numbers (e.g. "800-123-1234"), dates (e.g. "10/1990"), and other special queries (e.g. "u2" and "f(x)", the musical bands) are very popular non-math queries that are hard to differentiate from math queries. Hence, determining math queries among a collection of arbitrary queries is not an easy task, and relying only on the structure of queries does not completely address it.

In this paper, we propose a novel approach to address the problem of identifying math queries among a collection of arbitrary queries submitted to a general-purpose search engine. Instead of relying on the structure of queries, we use information in the search logs to recognize classes of math queries. Using this technique, we can determine how likely a collection of similar queries is math, regardless of their structures.

We also study the effect of answering math queries on the behaviour of users, and determine how well such answers meet their information needs. As we mentioned ear-



convert 10 meters to feet

About 31,500,000 results (0.25 seconds)

10 meters = 32.808399 feet
More about calculator.

convert meters to feet - Metric America
www.metricamerica.com/**convert**%20**meters**%20to%20**feet**.
convert meters to feet in America. **...** 1 **meter** = 3.2808399
inches 1 kilometer = 0 .621 mile **...** (1000 g). ALL SIMPLY R

(a)

sin(pi/3)^2

About 3,030,000 results (0.24 seconds)

sin(pi / 3)^2 = 0.75
More about calculator.

How do I evaluate **sin(pi/3)** without using a calculat
answers.yahoo.com › ... › Mathematics
10 Feb 2008 – So, sin(90 degrees) = sin(90*pi/180) = sin(pi/**2**).

(b)

Figure 2: Google's result page for (a)"convert 10 meters to feet", and (b) "sin(pi/3)ˆ2"

lier, some queries might be mistakenly identified as math queries, and hence a math answer to such queries is normally not useful. Moreover, some math queries are ambiguous and answers to them might be different from what the user expects. For example "20ˆ1/2" could be interpreted as "20ˆ(1/2)" or "(20ˆ1)/2", and "20 K" can be interpreted as "20 Kelvin", "20 thousand", or "20 * Boltzmann constant". Therefore, it is still necessary to evaluate the usefulness of answers. Math answers do not contain links, therefore such evaluation is different from the traditional evaluation problem where the search result is assumed to contain links that could be clicked on. In this paper we describe two metrics that, as implicit feedback from users, can reflect the usefulness of results.

The rest of this paper is organized as follows. We overview the related work in Section 2. In Section 3 we propose an algorithm for clustering and recognizing math queries, and evaluate it through empirical and statistical analysis. We study the behaviour of users when math answers are shown in the result page, and describe metrics for evaluating the quality of such answers in Section 4. We finally conclude the paper.

2. RELATED WORK

Chilton et al. [4] study various types of queries that could be directly answered in the result page, and propose various parameters that can be used as implicit feedback from users to evaluate the quality of results. To define some of the parameters, it is assumed that answers contain links. For example an answer to a query about flights between two cities contains prices and dates, and links to travel agency websites. Math answers do not contain links, and the only parameters that can be applied are the number of times a user issues queries of the same type, and the decrease in the number of clicks on ranked documents (if a list of documents in addition to an answer is shown in the result page). The former is based on interpreting the repeat triggering of an answer by the same user as a sign of satisfaction, and the latter on interpreting clicks on links outside an answer as a sign of dissatisfaction with the answer. A study by Castillo et al. [3] also confirms that short answers reduce the average click rate. Stamou et al. [14, 15] study users satisfaction when they do not click on any link for a keyword query. They propose metrics such as the time spent on a result page combined with the scrolling information and terminological overlap between consecutive queries. Typically, scrolling information is not available to a search engine. Also because math queries tend to contain none or very few words, their terminological overlap is not as indicative as in the case of keyword queries. Hassan et al [5] propose an approach for evaluating search engines that is based on modelling sequences of users interactions with the engine.

Mathematics retrieval is an area of research where a query is assumed to be a mathematical expression, and the problem is to retrieve documents, or other units of data, that contain similar expressions to the query [7, 9]. This problem is inherently different from the problem that we address in this paper. We assume answering math queries rather than matching mathematical expressions to the query and retrieving documents accordingly, and we focus on issues such as recognizing math queries from other queries and evaluating the usefulness of answers.

The problem of identifying classes of math queries is related to the problems of query similarity and query clustering. There are many proposals for calculating the similarity of web queries and clustering them. Shen et al. [13] propose an algorithm for web query classification based on enriching queries using search snippets and also click-through data in the search logs. Wen et al. [18] propose a query clustering algorithm using query logs. In this approach, the similarity of two queries is defined based on the common documents that users selected for them. In the case of math queries, there is no link to click on and no search snippet, so these approaches are not applicable. Bordino et al. [2] propose an algorithm for calculating the similarity of queries using query logs. They construct a graph with each query as a node. There is a weighted link between every two nodes that represent the number of sessions that contain both queries. Next, the graph is projected to a Euclidean space and cosine similarity between queries is calculated. Math queries tend to contain many numbers and symbols that do not significantly contribute in the semantics of the query. This affects the usefulness of this approach for math queries because for example changing the value of a number results in adding a new node to the graph.

3. RECOGNIZING MATH QUERIES

In this section we propose an algorithm for automatically identifying math queries among a collection of arbitrary queries taken from the search logs. It allows us to determine what fraction of math queries are correctly recognized and answered by a search engine, and what fraction of the queries recognized as math are actually math queries. It also allows discovering classes of math queries that are not handled by the engine,and enhancing it by developing appropriate modules to recognize and process them.

In many cases, the intention of a user cannot be inferred by merely looking at the query [5]. For example while a user might mean "log(100)" from "lg 100", another user might issue this query to search for LG-100 cellphones. Also, as we mentioned earlier, a complete set of grammars that describes the structures of the entire range of math queries is not available. Hence, we need extra information to correctly recognize math queries.

Search logs contain valuable information that could reflect the intention of a user from issuing a query. If queries in a cluster are mostly asked by users who tend to ask math queries, it is a sign that they are probably math queries. Therefore, we present an algorithm to predict if a user tends to ask math queries within a search session.

According to the search logs, we observe that when a user asks math queries in a session, she usually tends to ask more math queries within the same session. We also observe that the majority of math queries contain numbers, math symbols, or math keywords. It allows us to easily identify queries that are potentially math. Note that many queries that are potentially math are actually not math queries, but almost all math queries are potentially math. The last two observations imply that sessions within which a user tends to ask math queries, should contain a rather large fraction of potentially math queries. After identifying such sessions, we can identify clusters of queries that are mostly asked when a user tends to ask math queries. In Section 3.3 we explain details of this algorithm, and prove its correctness.

Because numbers and symbols are common in math queries and their values contribute less in the semantics of them, identifying classes of math queries rather than individual math queries is more useful. Therefore, a clustering algorithm that groups similar queries together is necessary. Accordingly, our algorithm consists of two main parts: clustering queries, and recognizing clusters that mainly consist of math queries.

3.1 Marking potentially math queries

We first mark queries as potentially math queries, or non-math queries. Ideally, all math queries are marked as potentially math, but there might exist some, or many, potentially math queries that are not math. To ensure most math queries are marked as potentially math, we apply the following simple heuristic. Any query that consists of numbers, alphabetic symbols (i.e. a string of length at most 2), non-alphanumeric symbols, or a predefined set of math keywords is marked as potentially math. Among the set of queries that are marked this way, many queries such as part numbers, phone numbers, and dates are not math queries.

We apply the clustering algorithm described in the next section only on potentially math queries, and in the remainder of this section, by a query we mean a potentially math query.

3.2 Clustering queries

Unlike many text queries, two similar math queries might share very few symbols in common. For example "12*120+20" and "33*47+9" are very similar queries even though only '*' and '+' are in common between them. As we mentioned earlier, math queries are typically short, and numbers, alphabetical characters, and math symbols appear frequently within them. Although numbers and variables are common in math queries, their values usually do not significantly affect their meanings (e.g. "12y+1 = 0" and "150x+10 = 4" are similar).

In order to compare queries, we first transform them into canonical forms by performing various normalizations. We replace numbers and characters representing variables with special strings, "NUM" and "VAR". For example "12 y + 1 = 0" is transformed into "NUM VAR + NUM = NUM". Plus and minus operators are mathematically similar, therefore we replace both with "PLMN" ("12 - 2 + 4" is transformed into "NUM PLMN NUM PLMN NUM"). In our implementation we also take into account the type of a number (e.g. floating point or integer), and also distinguish large and small integer numbers, but for the ease of explanation we do not get into such details . We replace other parts of a query such as units, operators and geometry objects with their equivalent canonical forms (e.g. "10 m to ft" and "10 meter to foot" are both transformed into "NUM UNITS to UNITS" and "calculate the area of a 22x34 rectangle" into "calculate the GFUNC of GOBJ").

There are some words that appear in many math queries, that similar to stop words in keyword queries, do not affect the results (e.g. "calculate" in "calculate 12*120+20"). We remove such words from a query.

After doing similar transformations, we next consider repeating patterns within a query. In some math queries, a part of the query can repeat arbitrarily often without changing the meaning of the query. For example "12+3+27+39" and "12+2" are similar queries. There are various proposals for inferring regular expressions from a sample [8] or finding repeating patterns within a sequence of items [8, 11]. We choose the algorithm explained in [11] to recognize such repeating patterns, and replace them with a single occurrence of them (e.g "NUM+NUM+NUM" is transformed into "NUM+NUM").

Finally, we remove all space and separator symbols.

Next, we compare two queries in canonical forms using 3-grams and cosine similarity [17] as follows. Each query is transformed into a vector of 3-grams. Assume Q_1 and Q_2 are two queries in canonical forms represented as vectors of 3-grams:

$$Sim(Q_1, Q_2) = \frac{Q_1.Q_2}{|Q_1||Q_2|}.$$

Even though the above similarity function treats expressions as plain strings which causes some math semantics to be lost, it can be calculated efficiently and according to our experiments it works well in practice in most cases. As a part of our future work we consider defining a more sophisticated similarity function that better captures the semantics of math expressions.

Based on this similarity function, we use hierarchical agglomerative clustering [6] to form clusters that contain similar queries. The clustering algorithm starts by forming a cluster for each query. It then iteratively merges clusters whose similarity is greater than a threshold, θ. The similarity of two clusters is defined as follows:

$$sim(C_1, C_2) = min\{sim(a, b) | a \in C_1,\ b \in C_2\}.$$

The granularity of clusters can be controlled by adjusting the value of θ.

3.3 Identifying math clusters

So far, we explained how to mark queries that are potentially math. We also described a clustering algorithm that groups similar queries together. In this section we propose an algorithm to distinguish clusters that contain actually math queries from the ones that contain potentially math, but not actually math queries.

A search session is a sequence of queries issued by an individual with less than 30 minutes between sequential queries [4]. Therefore, if a user is idle for more than 30 minutes and then issues a query, a new session starts. An example of a search session is shown in Figure 3.

The main idea behind our algorithm is to identify sessions within which users tend to ask math queries. We call such sessions math-oriented sessions. If a cluster contains many queries that mostly appear in math-oriented sessions, it is an indication that this cluster probably consists mainly of math queries.

We observe that in practice, a rather large fraction of queries within a math-oriented session are math queries. In other words, potentially math queries that are actually math tend to appear frequently within a session while other potentially math queries, that are actually non-math, typically appear in isolation.

Based on this observation, for a session S, we define $M_d(S)$ as the fraction of distinct potentially math queries to the total number of distinct queries within S:

$$M_d(S) = \frac{|\{Q \in S | Q\ is\ PM\}|}{|\{Q \in\ S\}|}.$$

PM in the above equation stands for potentially math. We define $M_r(S)$ to be the fraction of potentially math queries within the session:

$$M_r(S) = \frac{\#\ of\ Q \in S | Q\ is\ PM}{|S|}.$$

Note that to calculate M_d, we only count distinct queries, so if the same query is asked more than once in a session, it is counted once in calculating M_d, while to calculate M_r, such repetitions are taken into account.

Then, we define $M(S) = M_d(S).M_r(S)$.
In the calculation of $M_d(S)$, repetitions of the same query within a session are ignored, and for $M_r(S)$ the diversity of queries does not matter. $M(S)$ results in something in between, and according to our experiments, it can better predict if a session is math-oriented. Math-oriented sessions tend to have higher number of potentially math queries, and hence a higher value of $M(S)$.

	P		
	Minimum	Average	Maximum
Non-math	0.04	0.12	0.27
Undecided	0.10	0.20	0.30
Math	0.18	0.33	0.51

Table 1: Minimum, average, and maximum value of P for math, non-math, and undecided clusters.

.

For a cluster C, we define $P(C)$ as follows:

$$P(C) = \frac{\sum\limits_{S_i \in \{S|(C \cap S) \neq \phi\}} (M(S_i))}{|\{S|(C \cap S) \neq \phi\}|}.$$

$P(C)$ represents the average value of M for all distinct sessions that contain at least one query from C. If a session contains more than one query from the same cluster, it is counted once only. Because short sessions are less informative, we only counted sessions that contain at least three distinct queries.

In the next section, the relationship between P and percentage of math queries in a cluster is statistically evaluated and modelled for predictive and descriptive purposes.

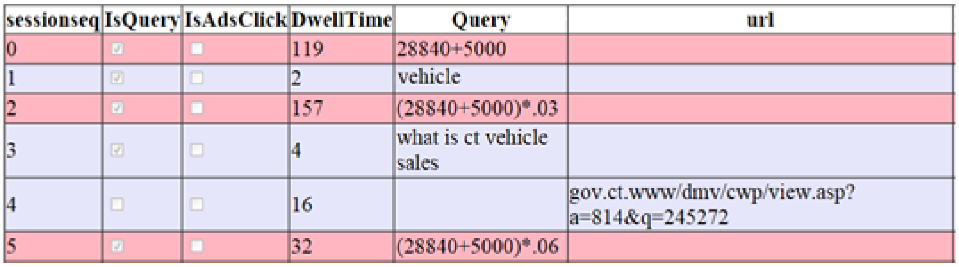
3.4 Data analysis and statistical modelling

In this section we present a statistical model that allows us to estimate the percentage of math queries in a cluster based on the calculated value of $P(C)$, and show that this model is statistically significant. The model is constructed using the data set generated from the following process.

We took a 1-percent sample from Bing's search logs from "4-26-2011" to "6-26-2011", and extracted queries with US-English language [1]. After marking the queries in this sample, we collected 71,800 potentially math queries. Applying our clustering algorithm with $\theta = 0.85$ resulted in 9,120 clusters. Among them, we picked a training set as follows. We chose 150 arbitrary clusters, and randomly selected 25 queries from each one. Then, we manually marked each selected query as math or non-math. In Figure 4, each cluster is represented with a red square. For each cluster, the x-axis represents the calculated value of P, and the y-axis shows the percentage of math queries.

The sample is summarized in Table 1. If more than 80% of queries in a cluster are math, we call it a math cluster, if less than 20% are math, we call it a non-math cluster, and otherwise we say it is undecided. As confirmed by t.student test [10], the average value of P for non-math clusters is significantly smaller than the similar value for math clusters at 99% confidence level. This suggests there is a statistical relation between the percentage of math queries and the value of P. In the remainder of this section, we further investigate

[1] The language of a query is determined by the language of the operating system of the client machine, that is available in search logs.



sessionseq	IsQuery	IsAdsClick	DwellTime	Query	url
0	☑	□	119	28840+5000	
1	☑	□	2	vehicle	
2	☑	□	157	(28840+5000)*.03	
3	☑	□	4	what is ct vehicle sales	
4	□	□	16		gov.ct.www/dmv/cwp/view.asp?a=814&q=245272
5	☑	□	32	(28840+5000)*.06	

Figure 3: An example of a session. Math queries are shown in pink.

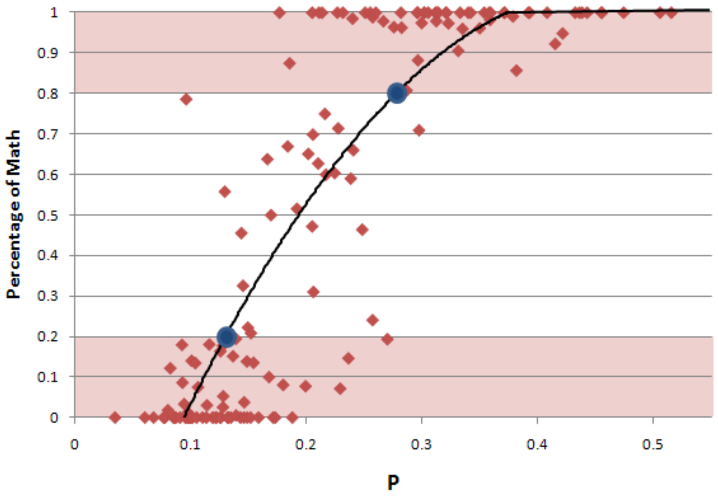


Figure 4: Change in the percentage of math queries within a cluster as P increases.

	$P \leq 0.11$	$0.11 < P < 0.32$	$P \geq 0.32$
Non-math	95%	40%	0%
Undecided	5%	36%	2%
Math	0%	24%	98%
total	100%	100%	100%

Table 2: Percentage of math and non-math clusters for various ranges of P.

this relation. .

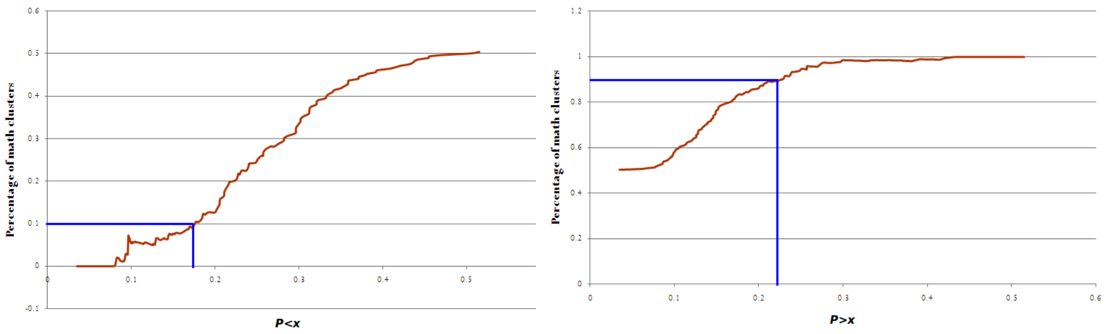
The relation between $P(C)$ and percentage of math queries in a cluster is modelled through a nonlinear regression. The choice of nonlinear regression curve is motivated by the nature of data as presented in figure 4. We chose the best curve from a selection of polynomial and logarithmic curves according to their R^2 values. In the context of statistical models, the coefficient of determination, R^2, represents the proportion of variability in the data that is explained by the model, and provides a measure of how well future outcomes are likely to be predicted by the model [16]. According to the least square estimation method [12], the best fit is the following polynomial function (the black curve in figure 4):

$$T(C) = \begin{cases} -8.14P(C)^2 + 7.38P(C) - 0.62 & for\ 0 \leq P(C) \leq 0.45 \\ 1 & for\ P(C) > 0.45 \end{cases} \quad (1)$$

In the above equation, $T(C)$ represents the percentage of math queries in cluster C. All coefficients are significant at 95% confidence level. The model's R^2 is 0.759. That is, the above relationship has captured approximately 76% of the variability within the data. In addition, the analysis of variance (ANOVA) [1] for this model results in a highly significant regression (F-Value:225.54 with 2 and 143 degrees of freedom) with prediction error bond of 0.047.

The above model allows us to estimate the proper thresholds on the value of $P(C)$ to predict if a cluster is math or non-math. These thresholds are the values of $P(C)$ that result in $T(C) = 0.2$ and $T(C) = 0.8$ according to equation 1. As highlighted with blue circles in figure 4, the thresholds for math ($T(C) \geq 0.8$) and non-math ($T(C) \leq 0.2$) clusters are 0.11 and 0.32 respectively. The prediction power of these thresholds is demonstrated in Table 2. According to this table, 95% of clusters with $P(C) \leq 0.11$ are non-math, 5% are undecided. Also, 98% of clusters with $P(C) > 0.32$ are math, and 2% of them are undecided. In other words, the rate of correct guess is at least 95% for math and non-math clusters.

Similar data is presented in Figure 5-a, where the average percentage of math queries as a function of an upper-bound on the value of P is shown, e.g. on average, 10% of the queries in clusters with $P(C) \leq 0.17$ are math queries. Figure 5-b represents similar information, e.g. on average, 90%



(a) (b)

Figure 5: Average percentage of math queries in clusters with P (a)smaller, and (b)greater than a value.

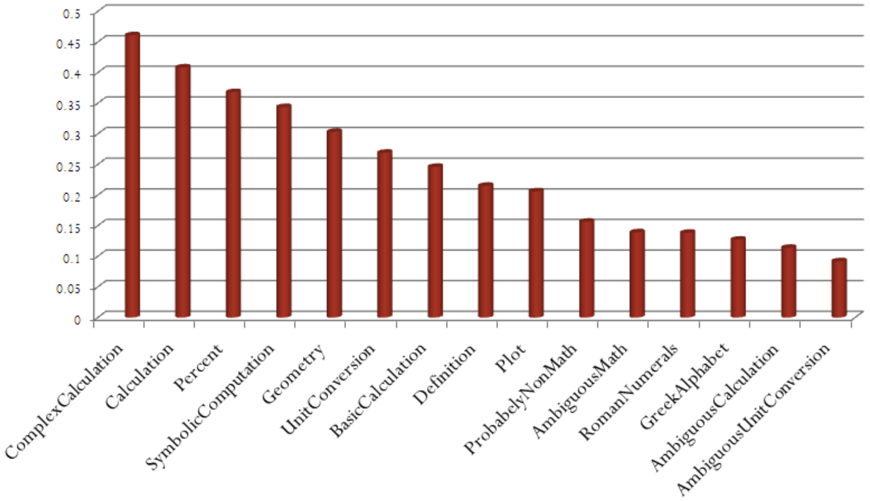


Figure 6: The average value of P for various types.

of queries in clusters with $P(C) \geq 0.22$ or more are math queries.

Note that the model is affected by the boundaries of the response and independent variables (i.e. $T(C)$ and $P(C)$) and is not powerful in recognizing small differences close to the boundaries. For example, consider two clusters, C_1 and C_2, with $T(C_1) = 0.95$ (i.e. 95% of queries in cluster C_1 are math), and $T(C_2) = 0.90$. Because $T(C_1)$ and $T(C_2)$ are close to 1 (the boundary value), our model does not differentiate them. It does not affect the usability of our model because such boundary values are treated equally in our application (e.g. both clusters with 90% and 95% of math queries are considered math clusters).

According to the above analysis, we conclude that our model can predict the percentage of math queries in a cluster C using the value of $P(C)$.

3.5 Further experiments

The clustering algorithm that we described in Section 3.2, is merely based on the structure of queries. However, some queries with the same template might have totally different meanings. For example, our clustering algorithm puts "9/11" and "146/23" in the same cluster, even though the first one probably means "ninth of September" while the second one means "146 divided by 23". In this section, we perform a manual clustering on a different data set to validate the model we derived in the previous section.

We extracted queries from Bing search logs for the first day of each month from January 2011 to July 2011. After marking queries, we randomly picked 1000 potentially math queries. We then manually assigned types to them, and put queries with the same types in the same cluster. For example "how many feet are in 10 meters" and "convert 10 litre/100km to mile/gallon" are both assigned type "unit-conversion" and hence are placed in the same cluster. Some types with their examples are listed in Table 3.

The average value of $P(C)$ for each cluster is presented in Figure 6. Complex-calculation and calculation queries have the highest value of P. Basic-calculation queries have a relatively lower values of P, which is consistent with the fact that some queries that belong to this class, although seem to be math, are actually a date or some other non-math queries, mistakenly marked by the human annotator. Ambiguous-unit-conversion and ambiguous calculations, have the lowest value of P, confirming the fact that in practice most such ambiguous queries are actually non-math.

Type	Example queries
Complex calculation	“square root of (10 choose 20)/log(20)”
Calculation	“(10+17.4)ˆ20/14”
Percent	“What percent of 14 is 10?” 12% of 127
Symbolic computation	“Solve yˆ2+y=0” “integral xˆ2+1 dx x from 1 to 10”
Geometry	“Volume of a 10-foot cube”
Unit conversion	“120 km/h to mph” “How many square feet are in ten square meters?”
Basic calculation	“145*12” “78/40”
Definition	“Binomial coefficient for 6, 4”
Plot	“x+1 graph”
Probably non-math	“9/11” “1834-1876”
Ambiguous math	“12a-1”
Roman numerals	“8 in roman numerals” “II”
Greek alphabet	“omega 3”
Ambiguous calculation	“3*-4*”
Ambiguous unit conversion	“12h”

Table 3: Example queries for each type.

3.6 Discussion

In the previous sections we showed that our model can correctly predict if a group of queries is math or non-math in most cases. Our prediction might be wrong when we cannot correctly guess if a session is math-oriented, or when a cluster contains queries of different natures (i.e. a mixture of math and non-math queries). To address the first issue, we can use other heuristics that help to better identify math-oriented sessions. For example, clicking on web pages with math contents is an indication that a user is interested in math, and hence the session is more likely to be math-oriented. It requires an algorithm to automatically determine if a web-page has math contents, and possibly a predefined list of math web sites. This is out of the scope of this paper, but is one direction of our future work.

We mentioned earlier that in some cases we cannot infer the intention of a user by just looking at the query. However, the clustering algorithm we described in Section 3.2 relies only on the structure of queries. It might result in clusters that contain a heterogeneous collection of queries. As pointed out in Section 3.5, the technique we describe for recognizing math clusters can be combined with any clustering technique. Our model can better predict the percentage of math queries in clusters with more semantically similar queries. Therefore, our proposed clustering technique can be replaced by more sophisticated clustering algorithms to enhance the results.

It might be argued that our model is biased in favour of math queries that are already answered by Bing. In our experiments, many queries that are identified as math by our model, and manually confirmed to be math queries, are not currently answered. For example according to Figure 6, our model can correctly predict the percentage of math queries for types such as complex-calculation and geometry, while such queries are not currently answered properly by Bing.

4. USEFULNESS OF ANSWERS

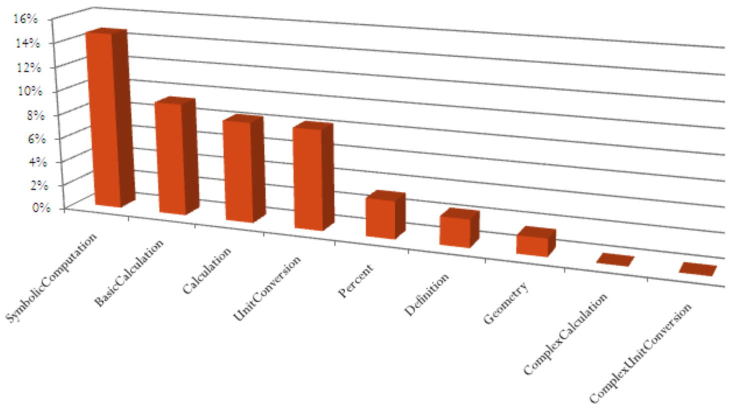


Figure 7: Fraction of users who issued similar queries in at least three different sessions.

So far, we are able to predict if a collection of queries is math or not. It allows us to automatically identify math clusters, and discover the clusters of math queries that are not currently answered. Next, we evaluate the usefulness of given answers. We also study how answering math queries affects the behaviour of users.

Traditionally, most IR evaluation techniques consider clicking on search results as implicit feedback from users. However, math answers do not contain links, so such evaluation techniques cannot be directly applied to math answers.

A recent study shows that repeat triggering of answers of the same type by the same user is a sign of satisfaction [4].

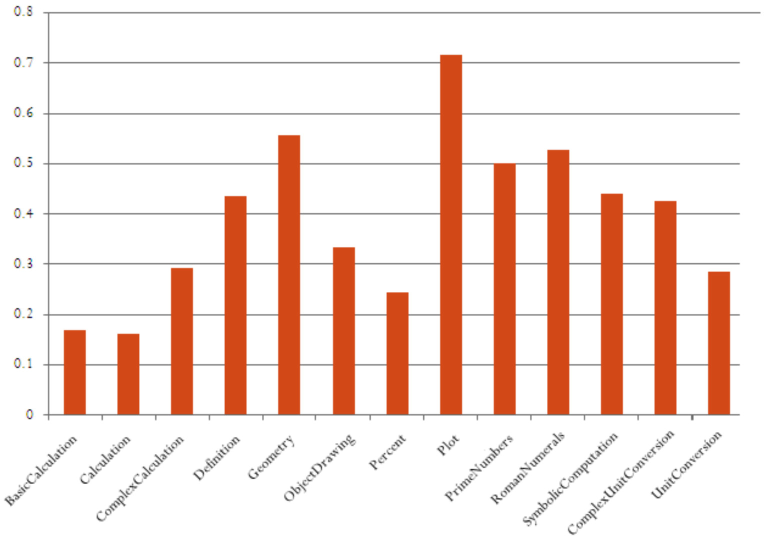


Figure 8: The value of d_U for various types.

Based on this observation, for a specific cluster, we define a returning user to be an individual who issues one or more queries of this cluster in at least three distinct sessions. We also define the returning user rate for a cluster as the ratio of the number of returning users to the number of distinct users who asked a query of this cluster at least once. In Figure 7, the returning user rate for various classes of queries is shown (each type represents a cluster as discussed in Section 3.6). Complex-unit-conversion (e.g. "what is the area of a 10*20m rectangle in sq feet?") and complex-calculation queries are not currently answered, so they have the lowest average ratio of returning users. Many symbolic-calculation (e.g. "solve xˆ2+x+1=0") and basic calculation queries are correctly answered, hence users who issue such queries are probably happy with the results and ask them rather often.

Some evaluation algorithms represent a session as a sequence of genes (i.e. symbols that encode various types of user's interactions with the engine) [3, 5]. Similarly, we represent a session by a sequence as follows. The start and end of the session are represented with 'S' and 'E'. If a query is a math query, we represent that with its cluster-id (the identifier of the cluster that the query belongs to), otherwise we represent it with Q. A click is represented by 'C'. Note that although a math answer does not contain links, but for many math queries a ranked list of documents is also shown in the result page, and a user might click on them (e.g. Fig 3). For example the session shown in Figure 3 is represented with the following sequence:

$$S[1310004859]Q[256658805]QC[256658805]E$$

If a math query is answered, but a user clicks on one of the retrieved documents, it implies that the answer could not sufficiently satisfy her needs. We also observe that a consecutive sequence of queries of the same cluster followed by a click is a sign that a user repeatedly modifies a query to get an answer, and after giving up, clicks on a result. Therefore, given a cluster U and session I, we define the following metric:

$$d_{U,I} = \frac{\#\ of\ \text{``}[[id(U)]]C\text{''}\, inI}{\#\ of\ \text{``}[[id(U)]]\text{''}\, inI}$$

$[[id(U)]]$ is a maximal consecutive sequence of $[id(U)]$s within the session's interaction sequence. For example assume the interaction sequence of a session is:
"$S\underbrace{[100][100][100]}\ C[200]E$".

$[[100]]$ is marked with an under-brace, and $d_{U,I} = \frac{1}{2}$. Note that a simple click-rate metric results in $\frac{1}{4}$, and it assumes the answers to the first two queries of cluster [100] are good and only the third query is followed by a click and hence not correctly answered.

For each cluster U, we calculate d_U, the average of $d_{U,I}$ for all distinct sessions within which at least one query from U is asked. In Figure 8, d_U for various types is shown. Many plot and geometry queries are not currently answered, and they have the highest click rates. Not surprisingly, calculation and unit conversion queries have relatively low click rates.

From May 1^{st} 2011 to Jun 24^{th} 2011 , the average d_U for all math clusters in 1-week periods is shown in Figure 9 (each week is represented by its last day). We observe a decline as new features are added to our math answering system over time.

According to our experiments, we can conclude that the ratio of returning users, and click rate are indicative measures for estimating the usefulness of math answers.

5. CONCLUSIONS AND FUTURE WORK

For some queries, providing a user with a short answer as well as a ranked list of documents can better satisfy her information needs. In this paper we studied math queries, a class of such queries for which an answer should be calculated rather than being looked up in a database. We described techniques for calculating the similarity of math queries and clustering them. We then proposed an algorithm for identifying math queries among a collection of arbitrary queries using the information in Bing's search logs,

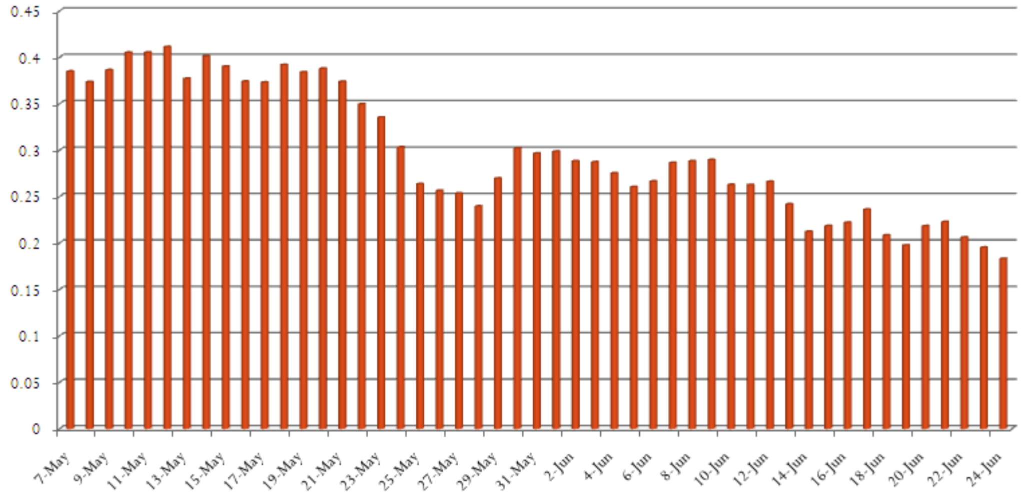


Figure 9: The average value of d_U over time.

and discussed its accuracy through statistical and empirical analysis. The advantage of this algorithm is that it works well in cases where the structure of a group of math queries in not known in advance, or where a non-math query looks like a math query. Finally, we investigated how answering a math query correctly, affects the behaviour of users. We described two parameters that, as implicit feedback from users, indicate how satisfied they are with the answers. To the best of our knowledge, the problems that we addressed in this paper have not been previously studied.

As a part of our future work, we wish to extend our algorithm to recognize math queries on the query processing time. It helps to decide if an answer should be triggered for an ambiguous query. For example if “lg 100” should be treated as a math query (“log(100)”), or it should be interpreted as “LG’s cell phone model 100”.

We mentioned earlier that other information can be deployed to enhance the results of our math detection technique. For example clicking on pages with math contents is an indication that a user is interested in math, and is more likely to ask math queries. Therefore, we wish to deploy such information to enhance our algorithm. Another direction of our future work is to study cases where a math query is mixed with many noisy keywords (e.g. “what is the area of a 20*22 meter land located in France in square feet?”).

6. REFERENCES

[1] F. J. Anscombe. The validity of comparative experiments. In *Journal of the Royal Statistical Society*, pages 181–211, 1948.

[2] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *SIGIR*, pages 515–522, 2010.

[3] C. Castillo, A. Gionis, R. Lempel, and Y. Maarek. When no clicks are good news. In *SIGIR*, 2010.

[4] L. B. Chilton and J. Teevan. Addressing people’s information needs directly in a web search result page. In *WWW*, pages 27–36, 2011.

[5] A. Hassan, R. Jones, and K. L. Klinkner. Beyond dcg: user behavior as a predictor of a successful search. In *WSDM*, pages 221–230, 2010.

[6] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[7] S. Kamali and F. W. Tompa. A new mathematics retrieval system. In *CIKM*, pages 1413–1416, 2010.

[8] S. Kamali and F. W. Tompa. Grammar inference for web documents. In *WebDB*, 2011.

[9] M. Kohlhase and I. A. SÂÿucan. A search engine for mathematical formulae. In *Artificial Intelligence and Symbolic Computation*, pages 241–253. Springer, 2006.

[10] P. Lewicki and T. Hill. *Statistics : Methods and Applications*. StatSoft, 2006.

[11] B. Liu and Y. Zhai. NET - a system for extracting web data from flat and nested data records. In *WISE*, pages 487–495, 2005.

[12] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, 2010.

[13] D. Shen, Y. Li, X. Li, and D. Zhou. Product query classification. In *CIKM*, pages 741–750, 2009.

[14] S. Stamou and E. N. Efthimiadis. Queries without clicks: Successful or failed searches. In *SIGIR Workshop on the Future of IR Evaluation*, pages 13–14, 2009.

[15] S. Stamou and E. N. Efthimiadis. Interpreting user inactivity on search results. In *ECIR*, pages 100–113, 2010.

[16] R. G. Steel and J. H. Torrie. *Principles and Procedures of Statistics*.

[17] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[18] J.-R. Wen and H. Zhang. Query clustering in the web context. In *Clustering and Information Retrieval*, pages 195–226. 2003.