Roles of Math Search in Mathematics

Abdou Youssef[⋆] [⋆⋆]

Department of Computer Science, The George Washington University, Washington
DC 20052, USA,
ayoussef@gwu.edu,
WWW home page: http://www.seas.gwu.edu/~ayoussef/

Abstract. Math-aware fine-grain search is expected to be widely available. A key question is what roles it can play in mathematics. It will be argued that, besides finding information, math search can help advance and manage mathematical knowledge. This paper will present the short-term goals and state of the art of math-aware fine-grain search. Afterwards, it will focus on how math search can help advance and manage mathematical knowledge, and discuss what needs to be done to fulfill those roles, emphasizing two key components. The first is similarity search, and how it applies to (1) discovering and drawing upon connections between different fields, and (2) proof development. The second is math metadata, which math search will surely encourage and benefit from, and which will be pivotal to mathematical knowledge management.

1 Introduction

Since the advent of the Worldwide Web, serious efforts have been undertaken to create digital libraries of mathematical contents, and to develop languages, tools, and systems for faster dissemination and processing of such contents [1, 3, 5, 11–14, 16, 20, 21, 28, 31–33, 36, 21, 22]. For digital libraries of mathematics to serve their purpose fully, users need to be able to search easily and effectively, especially for equations, functions, structures, proof patterns, and other kinds of fine-grained mathematical constructs. Although text search has reached a high level of maturity [38, 34], mathematical expressions are highly symbolic and structured, and are not curreently searchable by the available text-search systems.

Field-based search systems are now widely deployed in several mathematics databases and by many mathematical content providers, such as Zentralblatt's ZMATH and MathDi [40, 23], the Jahrbuch Database [15], AMS's MathSCiNet [2], and various professional mathematical socities. These systems afford users more targeted search, such as search by author, subject, title, abstract, journal, series, reviewer, review text, and the like. Standard subject classifications, such

[⋆] This work was supported in part by The National Science Foundation Grant No. 0208818

[⋆⋆] Some of the work was done as part of the DLMF Project at the National Institute of Standards and Technology, USA

as MSC 2000 [30], helps to a considerable extent in focusing the search. Neverthless, like text-search systems, the current field-based search systems are neither meant nor able to provide access to fine-grained mathematical data.

It will probably be much more useful to the mathematical and scientific communities to have *math-aware fine-grain search* systems. The author has been conducting research and development on that kind of search [29, 39]; much of that effort is part of the Digital Library of Mathematical Functions (DLMF) project [18, 19, 29]. The immediate goal of the research on math search is to create math-aware systems that (1) enable users to search not only for text, but also for fine-grain mathematical data, such as equations, functions, and structures; and (2) allow users to express math queries naturally and easily, using the notation and idiom of mathematicians and scientists.

Math-aware fine-grain search holds considerable promise for the short term and the long term. For the short term, it will help users fulfill momentary information needs. Whenever a user needs information about a specific mathematical item, s/he formulates and submits a query to the search system, which processes the query and returns a number of matching hits, ranked by relevance (or by some other user-specified criteria). The user will then browse through the returned hits, looking for the truly relevant ones which satify the need that prompted the search. At times, the user may have to refine their queries and repeat the search cycle. However, it is expected that the math-awareness of the search system is likely to identify much more relevant matches, and the fine-grain nature of the search leads to hits that point to **small-size** units of information. These two outcomes will greatly reduce the amount of time a user spends on searching and browsing through hits to find what is needed, and thus enable the user to return quickly to the main task at hand.

For the long term, math-aware fine-grain search holds promises that have potentially broader scope and greater impact. Specifically, it will be argued in this paper that such a search capability can contribute to the advancement and management of mathematical knowledge. For example, math-aware search can be used to find similarities between a piece of mathematics being developed, on the one hand, and proved theorems and well-developed theories in the same or different fields of mathematics, on the other hand, thus pointing the mathematician to fruitful methodological directions and interesting connections (note: two expressions or patterns are similar if some appropriately defined distance between their structures is below a certain threshold). Furthermore, through similarity search, it is possible to provide interactive *computer-aided proving* (CAP), either as a standalone system or as a complement to proof planning systems (e.g., λClam [6, 9] and Omega [4, 26, 27]). In the standalone mode, a CAP system can, by constant monitoring of an evolving proof or at the prompting of a user, automatically search for similar proof patterns, and thus periodically suggest to the mathematician relevant strategies, tactics, and/or logic rules that can be applied to further the proof. In the other mode, as a component of a proof planning system (PPS), math-aware fine-grain search can help the user first find initial plans of proof (of "similar" theorems), and later in the proof process find refinement tactics, all through ongoing search for similar plans and tactics against either a standalone knowledge base or Web-accessible math repositories of appropriately marked up contents and proof patterns. These and other potentialities of math-aware fine-grain search will be discussed later in the paper.

With regard to contributing to mathematical knowledge management, math-aware fine-grain search can help classify manuscripts. The search, in a semi- or fully-automated classification environment, can be used to find similarities and associations between different manuscripts. Using the similarities and associations, a librarian or a system can classify and characterize (with metadata) a previously uncategorized document, by borrowing the classes and descriptive metadata of the search-discovered similar documents. Furthermore, in a radical departure from current practices, this process of classification and metadata-enrichment can and sould be done at fine ganularity — at the level of equations, functions, structures, proof patterns, and the like. This can be done **by** and **for** math-aware fine-grain search.

It is evident from the above that similarity-search and metadata are fundamental to those envisioned long-term roles of math search, and to the symbiotic relationship between search, management, and advancement of mathematical knowledge. Similarity-search, a fairly developed area in data mining applications [37], is a new area in math search, and will be discussed later in this paper. As for metadata, international, professional, and academic efforts towards developing math metadata have been initiated, such as the MathNet project in Germany [25], and the activities of the International Mathematics Metadata Task Force (and its affiliated American task force) [24]. The planned metadata of those efforts seem to be at a coarse-grained level: at the level of books, articles and manuscripts. The benefits of such efforts towards improved access, dissemination, and management of mathematical knowledge, will be considerable. They will be even greater if the metadata is at a fine-grain level. Of course, providing metadata at any level, but especially at a fine-grain level, is a daunting task. Therefore, automatic generation of metadata is indispensable.

This paper will address the short-term and long-term objectives, roles and capabilities of math-aware fine-grain search. Specifically, the paper will identify the main aspects and pertinent issues, present the state of the art, and, where possible, outline approaches to follow.

2 Math-Aware Fine-Grain Search

This section will address the basic objectives, and issues, and state of the art of math-aware fine-grain search.

As a result of the work on and experience with the development of math search on the DLMF, the author has identified some key objectives that math search systems ought to meet, at least to a significant extent. The next subsection describes those objectives.

2.1 Basic Objectives of Math Search

1. **Math-awareness:** Much of the mathematical knowledge is embodied in mathematical symbols, elaborate notations, and structures of various levels of complexity. So for math-search systems to be effective, they have to recognize mathematical symbols and structures.
2. **A natural math-query language:** A math search system must provide an intuitive yet expressive math query language. Users in the mathematical and scientific communities should be able to express their queries in the same way as they would write other mathematical expressions, such as in a Latex-like syntax. Table 2.1 shows several examples of queries and describes the corresponding matching records.
3. **Fine granularity of searchable and retrievable information units:** With the vast and fast-increasing amount of mathematical knowledge available for electronic access, it is desirable to search for the most targeted information, be it an equation, an integral, a differential equation, a Fourier transform of a function, a definition, a graph, a theorem, a proof technique, etc. If such is the size (granularity) of what a user needs in a given situation, it would be a waste of the user's precious time to provide him/her a larger amount of information and expect him/her to sift through it to locate the relatively tiny piece of interest. Therefore, an important objective of math search is to aford users the ability to search for and retrieve fine-grain targets. (A *target* or *record* is any searchable and retrievable information unit in a database.)
4. **Perfect recall:** Recall is a standard metric in all search systems; the recall per query is defined to be the ratio of the number of relevant hits to the total number of relevant targets in the database. It is a universal objective of search to maximize recall. That is, every target that matches a query must be included in the hitlist.
5. **Perfect precision:** Like recall, precision is another performance metric of all search systems; the precision per query is defined to be the ratio of the number of relevant hits to the number of hits in the hitlist. Every attempt should be made to maximize precision. That is, every hit in the hitlist must match the query; the hitlist should not contain any false hits.
6. **Perfect relevance-ranking:** Ideally, if hit A is more relevant than hit B, then A should appear before B in the hitlist. In particular, the most relvant hit(s) must appear on top of the hitlist, or at least near the top. This objective is particularly pressing because of the very large and ever increasing number of potentials hits.
7. **Useful highlighting:** Highlighting within a retrieved target should be done in a way that informs and justifies to the user why the target matched, and which specific parts matched. For very fine-grained targets, such as an equation or a graph, highlighting is not so critical, but for large-grained targets such as an article or manuscript, highlighting is very desirable to help the user identify quickly the more relevant parts of the hit.
8. **Minimum hit-redundancy:** In systems where targets at different levels of granularity are available, some targets may be subsets of other targets, such

as a separately accessible equation that is a part of a separately accessible section. In such environments, redundant hits are possible. For example, if target A is a subset of target B, and if B matches a query only because A matches the query, then presenting both A and B as two separate hits in the hitlist constitutes redundancy. Hit A should be presented, and B should be left out. The objective is to eliminate redundancy. If that is too costly, an attempt must be made to reduce the effect of redundancy; for example, have hit B appear much later than hit A in the hitlist. Note that if the targets are disjoint, no redundancy should arise; redundant hits would be a reflection of poor system design.

Table 1. Examples of Queries

Query	**Matching Records**
sin^2 x+cos^2 x	Those containing the expression $\sin^2 x + \cos^2 x$
`J_n(z)=`	Those containing the fragment "$J_n(z) =$"
`Gamma(1/2)=`	Those containing "$\Gamma(\frac{1}{2}) =$", for the values of $\Gamma(\frac{1}{2})$
sqrt(Ai^2+Bi^2)	Those containing the expression $\sqrt{Ai^2 + Bi^2}$
^(x+2)	Those containing $x + 2$ as an exponent
int^infinity	Those containing $\int^\infty$
int (sin x)/x dx	Those containing $\int \frac{\sin x}{x} dx$
DeMoivre and cos (n theta)	Those containing both "DeMoivre" and $cos(n\theta)$
"Fourier transform" and spheroidal	Those showing Fourier transforms of spheroidal functions, in addition to those containing the terms "Fourier transform" and "spheroidal"
Ai and Bessel	Those showing connections between Airy Ai and Bessel functions, in addition to those containing the terms "Ai" and "Bessel"
Ai = BesselK	Ideally, those containing equations expressing the Airy Ai function in terms of the Bessel function K

2.2 Issues and Policy Decisions

In meeting those objectives, several fundamental issues must be faced and some policies for resolving them must be implemented. Here are some of the more important issues and challenging policy decisions that have to be handled.

- **Target definition and granularity**: The designer must define what should be a searchable and retrievable target, and decide on the appropriate granularities of targets.
- **Literal vs. abstract understanding and weighting of query terms**: Mathematics is rife with abstraction and levels of abstraction. As a simple

example, the name of a function argument is not to be taken literally, whereas the standard name of an elementary function or a special function should be taken literally. Another aspect is whether users can characterize rather than specify the terms that must occur in the matching targets. For example, can users enter "trigonometric" to stand for any term that is the name of a trigonometric function?

– **Whether to return mathematically equivalent hits, and to what extent**: Many a mathematical concept or expression can be expressed in several equivalent forms. The question is whether or not documents that do not contain a literal match of a query expression but contain an equivalent expression should be returned as hits. If the search is for "$\sin(\frac{\pi}{2}-x)$", should the system return documents containing the equivalent expression "$\cos x$"? How about if the query is "$\frac{1}{x}$" and a document contains "x^{-1}"? Some equivalences are so commonplace that users may wish them to be detected and matched in search, while other less familiar equivalences would cause confusion if detected and matched. The extent of equivalence-awareness in search is a serious design decision. Of course, the implementatioin of "deep-equivalence" awareness is a major task that requires sophisticated mathematical reasoning algorithms.
– **Determination of the intented meaning of a user's query**: There is considerable "overloading" of names and notation, i.e., the same symbol referring to different things in different contexts. For example, the zeta symbol (ζ) can refer to the Jacobi zeta function, the Weierstrass zeta function, the Riemann zeta function, or a generic symbol with no specific denotation. If a user includes zeta in a query and has a specific context in mind (e.g., number theory) but that context is not communicated in the query, the system will have no way of determining which zeta occurrences to match, or how best to rank the hits, creating a likely situation of high user dissatisfaction with the results.

All but the first point above involve the extremely challenging problem of determining the user's intent and wishes, without soliciting too much information per query from the user. Decision policies are needed in order to make "educated guesses" about the user's intent and wishes from the limited information provided in the query, and, accordingly, to determine what targets are truly relevant and how to relevance-rank the various hits. For more accurate assessment of relevance, the context of the search must be determined, such as the user's field and level of expertise, and the area of interest at the time of the search. It is worth noting that relevance is a relatively old, open question in the general field of text information retrieval (IR) [35], and the issue of context-based search is a current research topic with considerable interest in the IR community [17].

2.3 State of the Art of Math-Aware Fine-Grain Search

As mentioned in the Introduction, all search systems deployed by the current mathematics databases and mathematical content providers are conventional coarse-grain field-based text search systems with little math-awareness. In math-aware search, some work has started to appear. Recently, Guidi et al published papers on a math query language MathQL [12] and related searching techniques [11], both of which are for RDF metadata repositories, where RDF is the XML-based metadata markup language standard. The MathQL syntax is a markup style that is advanced in its expressive power, and requires the users to be advanced mathematicians. An earlier effort in math-aware fine-grain search is the work by Einwohner and Fateman [10], which was limited to integral-lookup.

The most recent work on math-aware fine-grain search is the work on the DLMF search [29, 39]. All the eight objectives presented in Subsection 2.1 have been met to a large extent. The resulting system, to be deployed in the near future, is fully math-aware and supports search and access to fine-grain targets such as equations, figures, tables, definitions, and named rules/theorems. It allows users to submit queries with Latex-like syntax. It achieves perfect precision and recall as far as term-occurrence search is concerned; also, through metadata enrichment, additional relevant hits are matched beyond literal occurence of terms. Relevance ranking is satisfactory, and is being improved. Small-grain targets (such as equations and figures) are highlighted when displayed within larger documents (such as sections). Finally, redundancy is greatly minimized, and when users restrict the search results to a specific type (such as equations or figures), no redundancy arises.

3 Objectives and Roles of Math Search in the Long Term

Beyond the conventional search for documents, it is envisioned that math search can fullfil higher-level and farther-reaching roles. This section will discuss some of those roles.

3.1 Discovery of Similarities between Fields

Research in an evolving new field (or sub-field) may discover preliminary patterns and laws that happen to be similar to those in older, more established fields. Early discovery of such similarities may suggest new patterns, laws, and properties, which are well-established in the older fields, to explore in the context of the new field. Also, proven useful methodologies in the older fields may apply to the new field fruitfully. The bridging and borrowing apply to both broad methodologies and specific proof techniques & patterns.

Math search can help in the discovery of such similarities — as long as the content repositories are well-formatted, adequately marked up, and accessible. Section 4 will discuss methods of discovering and measuring mathematical similarities. (Recall from the Introduction that two expressions or patterns are similar if some appropriately defined distance between their structures is below a certain threshold. That is, the two expressions/patterns are similar if their structures are identical or near-identical.)

Suffice it to say at this point that a search-driven technology of similarity-discovery is likely to increase productive interdisciplinary activities, not only between mathematicians of different specialties, but also between mathematicians and researchers in the natural and even social sciences. Indeed, it is often the case that scientists, who are in other disciplines than Mathematics and happen to be engaged in some mathematical work related to their disciplines, need to know what mathematical theories and knowledge can help them advance their fields, and which mathematicians are doing such work and can thus be invaluable collaborators. A math-similarity search capability can help such scientists locate relevant mathematical work and potential collaborators.

3.2 Computer-Aided Proving

A second major role that math-aware fine-grain search can play is computer-aided proving (CAP). That can take at least two shapes: (1) a straightforward online computer-aided proving (O-CAP) role, and (2) a more elaborate interactive real-time computer-aided-proving (R-CAP) role. Both are discussed next.

Online computer-aided-proving A user engaged in developing a proof for a theorem can, at various junctures of the proof development, submit expressions and Logical patterns (from the evolving proof) as queries. Matches may contain "identical" or similar proofs that are complete and valid; such proofs can then be mimicked, or learned from, to complete the proof at hand in an analogy-driven fashion. Also, atomic entities, expressions, and possible patterns from the premises (or conclusions) of the to-be-proved theorem can be submitted as queries, to search for similar theorems; the corresponding proofs may serve as a good aid to the proof at hand. Note that this O-CAP functionality is easy to have and use, for it is nothing more than straightforward math-aware fine-grain search.

Real-time computer-aided-proving This is similar to O-CAP except that no explicit queries need be formulated and submitted by the user. Instead, a software system will, in the background and during the course of a proof-development, carry out the following steps:

1. monitor the evolving proof;
2. formulate & submit queries (from the expressions and logical patterns present in the partial proof);
3. search for similar expressions and logical patterns
4. evaluate, rank, and distill the returned matches; the distilling involves
   - identifying the known properties of entities (e.g., functions and operators) and of the premises/hypotheses; the entities and premises are those that are in the theorem or in the emerging proof.

  - identifying intermediate theorems/lemmas, as when the query consists of premises (from the original theorem or the emerging proof), and the matching hit is a theorem with the same premises; the conclusions of those matching theorems, and possibly their proofs, as well as bibliographic references to them, will be among the distilled materials presented to the user.

5. report the distilled results in real time as suggested directions (tactics) and intermediate sub-conclusions to the mathematician that is developing the proof;
6. repeat this cycle (steps 1-5) throughout the proof development, until the end of proof.

Note that in Step 3 of the R-CAP cycle, as well as in O-CAP, the search can be conducted not only against a local knowledge base, but also against all kinds of math repositories. For this to work, the math repositories must be well-formatted, adequately marked up, and indexed for searching. Such repositories are growing in size and number. They include: the DLMF [18, 19, 29]; MBase [16]; Mizar (at mizar.org); and so on.

An R-CAP implementation can be very much like *integrated development environments* (IDEs), which are very widely used by software developers in the computer science community. (Good Latex editors are small instances of IDEs). In an IDE, static and locational dynamic menus are available. The static menus offer fixed services and functionalities. Dynamic menus are menus whose items change depending on the context, and are populated by search systems working in the background; those menus pop up when the user mouses over certain words or commands in the file, or when the user types up the first few characters of certain patterns. An R-CAP IDE can behave in similar ways by popping up dynamic menus containing suggestions for new logical patterns/tactics/rules to follow, and those suggestions vary depending on where in the proof the user is, and what premises and intermediate conclusions have been put in the proof. The suggestions in the dynamic menus will be constantly gathered and updated by the R-CAP math search, which is working in the background. The search items that populate the dynamic menus in typical IDEs are usually obtained from search against an internal database as well as against the opened file. In an R-CAP IDE, however, the search can be extended beyond a local knowledge base (KB) and the opened file, to include Web-accessible knowledge bases; the user of the IDE would also have the configuration option of specifying which specific external KBs to make use of.

CAP as presented above bears a strong relation to proof assistants and proof planning systems in particular. Proof planning was introduced by Alan Bundy for inductive theorem proving [6, 7], and was implemented in the systems Clam λClam, and IsaPlaner [8]. The Omega system [4, 26, 27] extended Clam's proof-planning paradigm to knowledge-based proof planning. Proof planning systems start with an abstract-level proof plan, and then "carry out" the abstract-level plan, interactively (with the user) and recursively when needed, by expanding the steps into concrete sequences of logical steps.

Therefore, the proof-planning approach (of Omega and Clam) to proving is primarily top down: from an abstract proof-plan to a final detailed proof. The search-driven R-CAP approach, described above, is fundamentally an incremental, bottom-up approach, driven mainly by the direction that the mathematician is taking in the proof, but at the same time helping the mathematician to further that direction along, or suggesting alternative tactics and patterns as a result of similarity search.

3.3 Learning Aid

In addition to its research-furthering roles, math search can be used by math & science educators and students for educational purposes: finding what they need, and learning from what they find. It is an obvious and natural role of any search system.

In the context of math education, however, some issues arise. One important issue is the relationship between the granularity of the retrieved information, on the one hand, and the information need and the educational level of the user, on the other hand. For example, if a physicist is seeking the value of an integral or the general solution to a specific differential equation, the search results should be at the level of an equation, rather than the title of a book about the subject. Likewise, if a novice wishes to learn about number theory, the search results should be books and perhaps articles about the subject, rather than stand-alone equations about the Riemann ζ function or the Euler φ function. The notions of relevance and context-based search mentioned earlier are pertinent here.

Another issue is how best to integrate math-aware search into math learning systems in a synergistic fashion. Like many of the ideas discussed in this paper, this integration topic is in its infancy, and will require considerable research.

3.4 Routing

Routing is the process of informing users (or subscribers) of the latest information that match a pre-determined query specified by the user, as soon as the information becomes available. Math search can be used to stream to a mathematician all articles and manuscripts that match the mathematician's pre-specified query (or queries) whenever and as soon as the information becomes available. The source of the information can be professional societies, publishers, or researchers posting their manuscripts on their institutions's Web sites. The system(s) to do the routing can be centralized systems on the information providers' Web servers, or federated systems that periodically "crawl" the Web (or at least certain specific sites) searching for newly posted information. Either way, math search must be a central component of the routing system, and the posted information must be formatted and marked up adequately, and indexed, in order for the background searching to take place and the search results to be routed to the appropriate users, each according to his or her pre-specified queries.

4 Methods for Discovering and Measuring Similarity

As seen throughtout the paper, similarity search is useful in many contexts. It is referred to sometimes as approximate search or fuzzy search. Before one can proceed further, a formal definition of similarity is called for.

Definition 1. *Given a distance metric d in the "space" of mathematical expressions or patterns, and given a threshold h, two expressions or patterns E_1 and E_2 are said to be* h-similar *if $d(E_1, E_2) < h$.*

Two remarks must be made. First, the distance d need not be a distance in the strict topological sense, nor the "space" of expressions or patterns need necessarily be a topological space. Rather, d should satisfy the two properties

$$d(E_1, E_2) \Leftrightarrow E_1 = E_2,$$

$$d(E_1, E_2) = d(E_2, E_1)$$

But the triangle inequality is not essential.

Second, the actual definition of the distance d must capture, to the extent possible, the intuitive subjective notion of (dis)similarity between expressions or patterns. For example, $x^2 + y^2$ and $\cos^2\theta + \sin^2\theta$ are intuitively similar expressions, whereas $x^2 + y^2$ and $\int x dx$ are quite dissimilar. Also, d must capture comparative information about similarity, that is, if E_1 is more similar to E_2 than F_1 is to F_2, then we should have $d(E_1, E_2) < d(F_1, F_2)$. For example, one would expect that $d(x^2 + y^2, u^2 + v^2) < d(x^2 + y^2, x^2 + y)$.

Ideally, the distance d should be sensitive to the notion of different levels of abstraction. Specifically, if an expression E is an abstraction of another expression F, and F is in turn an abstraction of G, then one should have:

$$d(E, F) < d(E, G), \text{ and } d(F, G) < d(E, G).$$

Furthermore, if E and F are mathematically equivalent expressions but have different structures, as is the case for the two expressions $(x+1)^2$ and x^2+2x+1, then one would expect that $d(E, F) = 0$. This expectation, however, assumes that the similarity system incorporates the detection of logical equivalence and value-equavalence, which is a rather difficult problem of symbolic computation and automated mathematical reasoning. Therefore, for pragmatic reasons, one may wish to leave out the requirement that $d(E, F) = 0 \Leftrightarrow E \equiv F$, although preserving it can lead to much more interesting similarity results, at the cost of much more computationally intensive similarity measurement.

With all those considerations in mind, one approach to quantifying similarity (or distance) between mathematical expressions and patterns is by modeling expressions as parse trees with node labels that represent the names of functions/operators in the expression. Similarity (or distance) can then be defined using the structures and node labels of the trees. The more nodes with like-labels in the two trees, the more similar the trees. Also, the more sub-trees of identical structures that the two trees share in common, the more similarity there is. Similarity between the internal (non-leaf) nodes in the two trees is more important than similarity between the labels of the leaves in the two trees, because leaves often represent arbitrary variable names, while non-leaves represent essential operational and structural information of a math expression. Also, structural and label similarities higher up the two trees are often more important than those further down the trees, because the "fundamental" structure of a formula/expression is reflected more near the root of the parse tree. These differences in importance suggest weighted measures of similarity, where higher nodes and higher subtrees are assigned more weight than the lower ones.

The precise development of those ideas of quantifuing similarity, and the development of algorithms for measuring similarity (or distance), are subjects of ongoing research in the author's research group.

One final note is that once one has an adequate definition of similarity (or distance) and a good algorithm for computing the distance between two expressions/patterns, it is straightforward to incoporate the distance and its algorithm into math-aware fine-grain search systems for performing similarity search, at whatever level of desirable similarity (as specified by the threshold h).

5 Approaches for Generating Fine-Grained Metadata

In most application, metadata is generated manually. In fact, in many instances, metadata is extrinsic to the object being described, such as the date and journal of a publication; therefore, such metadata cannot be derived in any other way but manually. Fortunately, extrinsic metadata is small in size, and need be enetered at the coarse-grain level (i.e., at the level of books, articles and manuscripts). In the case when the metadata describes something intrinsic, such as the properties of a certain function, the properties may be so complex and intricate that only the author or a domain expert is in a position to unearth them and state them explicitly. For the sake of fine-grain search, the metadata will have to be entered at the level of equations, definitions, functions, proof patterns, and the like. The metadata must also to be marked up properly so search systems can make use of them. Both the metadata generation & entry, and the marking up, are time-intensive tasks that few authors would be willing to do. Therefore, it is preferable to automate the math-metadata generation process.

Most mathematical functions and concepts enjoy many properties and fall under a hierarchy of mathematical categories. To illustrate, assume that an equation (or math file) E has the cosine function "cos" in it. This function falls in the category of trigonometric functions, which is a subcategory of elementary functions, which in turn is a subcategory of special functions. It also enjoys the property of periodicity, among other things. Recall from earlier discussions that such properties are desirable to have as metadata. A user may wish to search for equations that have, among other things, periodic functions (or trigonometric functions, etc.). Clearly, even if the equation/file E does not contain explicitly any of those terms or phrases ("periodic functions" or "trigonometric functions"), E is a relevant object and should be returned as a hit. But without metadata, this is not possible.

We have developed a knowledge-based approach to generating metadata. First, a knowledge base was compiled, consisting of standard math functions and operators, on the one hand, and associated metadata on the other hand. Specifically, for every function and operator in the KB, the corresponding metadata is a set of descriptive phrases that name the properties that the function/construct enjoys, and the mathematical categories that the function/construct falls under. Afterwards, we developed algorithms that, for each equation and math expression, generate from the KB a combined list of the descriptive phrases of all the functions and constructs that occur in that equation/expression, and treat that combined list as metadata for that equation/expression. The approach was enhanced further through using the context of a math expression to derive additional metadata phrases. Specifically, the titles of the containing sections/subsections, the captions of the containing tables, and similar headers, can be used as sources of additional metadata. However, care must be taken when using context information, because every item of information in the context applies to every equation or expression in that context. For example, if the title of a subsection is "Fourier and Laplace Transforms", and the subsection contains several equations, some being Fourier transforms, and others Laplace trasnforms, then latching the entire title of the subsection to each equation in the subsection leads to inaccuaries.

A more powerfull approach to automated fine-grain-metadata generation derives "higher-order" metadata from the structure of an equation/expression, not just from the functions and constructs that occur in it. For example, the Fourier transform has a recognizable expression structure; an algorithm can be written to search for such structures in equations and expressions, and wherever found, associate the metadata phrase "Fourier transform" with the containing equations/expressions. This higher-order metadata generation requires (1) defining structural patterns (and their characteristics) for a number of mathematical constructs, such as the Fourier transform, the Laplace transform, (partial/ordinary) differential equations, recurrence relations, and so on; and (2) developing algorithms for recognizing such structural patterns in math expressions so corresponding metadata can be associated with them. One method for expression-structure recognition is to use expression parse trees, specifically their structures and the internal (i.e., operation) node labels, as templates for pattern recognition & classification.

Acknowledgements

The author thanks the DLMF team at the National Institute of Standards and Technology, USA, especially Drs. Bruce Miller, Dan Lozier, Ron Boisvert, Frank Olver, and Charles Clark, for the many discussions and their valuable comments on various aspects of math search.

References

1. The ActiveMath Project, http://www.mathweb.org/activemath/
2. MathSciNe. American Mathematical Society (AMS). http://www.ams.org/mathscinet
3. Asperti, A. et al: Mathematical Knowledge Management in HELM [Italy]. First International Workshop on Mathematical Knowledge Management, Schloss Hagenberg, Austria, September 24-26, 2001.
4. Benzmüller, C. et al: Ω: Towards a Mathematical Assistant, Conference on Automated Deduction, 1997.
5. Buchberger, B.: Mathematical Knowledge Management Using Theorema. First International Workshop on Mathematical Knowledge Management, Schloss Hagenberg, Austria, September 24-26, 2001.
6. Bundy, A.: The Use of Explicit Plans to Guide Inductive Proofs. R. Lusk and R. Overbeek (eds) Proceedings of the 9th Conference on Automated Deduction (CADE 9), Springer-Verlag, (1988) 111–120
7. Bundy, A.: Proof Planning. B. Drabble (ed) Proceedings of the 3rd International Conference on AI Planning Systems, (1996) 261–267.
8. Dixon, L. and Fleuriot, J. D.: IsaPlanner: A prototype proof planner in Isabelle. In F. Baader (ed), CADE 19, volume 2741 of LNCS, 279–283. Springer 2003.
9. Dixon, L., Jamnik, M., and Pollet, M.: Proof planning: Comparing Ωmega, λClam and Isa-Planner. In B. Bennett (ed) ARW 11, 50–52. University of Leeds, School of Computing, 2004. Held in Association with the AISB'04 Convention.
10. Einwohner, T. H. and Fateman, R.: Searching techniques for integral tables. International symposium on Symbolic and algebraic computation, ACM, 1995. (http://torte.cs.berkeley.edu:8010/tilu)
11. Guidi., F.: Searching and Retrieving in Content-based Repositories of Formal Mathematical Knowledge. Ph.D. Thesis in Computer Science, University of Bologna, March 2003. Technical report UBLCS 2003-06.
12. Guidi, F. and Schena, I.: A Query Language for a Metadata Framework about Mathematical Resources. The 2nd International Conf. Mathematical Knowledge Management, Bertinoro, Italy, Feb. 2003.
13. Hardin, T.: Mathematical Knowledge Management in FOC [France]. First International Workshop on Mathematical Knowledge Management, Schloss Hagenberg, Austria, September 24-26, 2001.
14. An Hypertextual Electronic Library of Mathematics, http://helm.cs.unibo.it/.
15. Jahrbuch Database. http://www.emis.de/MATH/JFM/JFM.html
16. Kohlhase, M.: MBase: Representing Knowledge and Context for the Integration of Mathematical Software Systems. Journal of Symbolic Computation 23:4 (2001) pp. 365 – 402
17. Leake, D. B. and Scherle, R.: Towards Context-Based Search Engine Selection. IUI01, January 14-17, 2001, Santa Fe, New Mexico, USA.
18. Lozier, D. W.: The DLMF Project: A New Initiative in Classical Special Functions. International Workshop on Special Functions - Asymptotics, Harmonic Analysis and Mathematical Physics. Hong Kong, June 21-25, 1999.
19. Lozier, D.W., Miller, B.R., and Saunders, B.V.: Design of a Digital Mathematical Library for Science, Technology and Education. Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries; IEEE ADL '99, Baltimore, Maryland , May 1999.
20. MathWeb.org, http://www.mathweb.org/

21. The OpenMath Standard, 1998, http://www.openmath.org/
22. MathML 2.0, a W3C Recommendation, October, 2003, http://www.w3.org/Math/
23. MathDi (Mathematics Didactics Database). http://www.emis.de/MATH/DI/
24. Mathematics Metadata. http://www.mathmetadata.org/
25. The MathNet Project. http://www.math-net.de/project/
26. Melis, E. and Siekmann, J.: Knowledge-Based Proof Planning. J. Artificial Intelligence, 1999.
27. Melis, E. and Meier, A.: Proof Planning with Multiple Strategies. First International Conference on Computational Logic, CL-2000 , 2000.
28. Melis, E. et al: ActiveMath: A Generic and Adaptive Web-Based Learning Environmen. Artifical Intelligence in Education,(12)4 Winter 2001.
29. Miller, B. and Youssef, A.: Technical Aspects of the Digital Library of Mathematical Functions. Annals of Mathematics and Artificial Intelligence, **Vol. 38** (2003) 121–136
30. Mathematical Subject Classification (MSC2000). American Mathematical Society. http://www.ams.org/msc/
31. MathWeb.org, http://www.mathweb.org/
32. The Omega Group, http://www.ags.uni-sb.de/õmega/
33. Rudnicki, P. and Trybulec, A.: Mathematical Knowledge Management in MIZAR. First International Workshop on Mathematical Knowledge Management, Schloss Hagenberg, Austria, September 24-26, 2001
34. Salton, G. and McGill, M. J.: Introduction to Modern Information Retrieval. McGraw Hill, New York (1993)
35. Saracevic, T.: Relevance: A Review of and a Framework for the Thinking on the Notion. Journal of the American Society of Information Science, **26(4)** (1975) 321–343
36. Theorist Interactive LiveMath, http://www.livemath.com/
37. Tan, P.-N., Steinbach, M., and Kumar, V. : Introduction to Data Mining. Addison-Wesley (2006)
38. Baeza-Yates, R. and Ribeiro-Neto, B.: Modern information retrieval, Reading, MA: Addison-Wesley, (1999)
39. Youssef, A.: Information Search And Retrieval of Mathematical Contents: Issues And Methods. The proceedings of the ISCA 14th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE-2005), July 20-22, 2005, Toronto, Canada.
40. Zentralblatt MATH database at European Mathematical Information Service (EMIS). http://www.emis.de/ZMATH/