Augmenting Presentation MathML for Search

Bruce R. Miller[1] and Abdou Youssef[2]

[1] Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD
bruce.miller@nist.gov
[2] Department of Computer Science, George Washington University, Washington, DC 20052
ayoussef@gwu.edu

Abstract. The ubiquity of text search is both a boon and bane for the quest for math search. A bane in that user's expectations are high regarding accuracy, in-context highlighting and similar features. Yet also a boon with the availability of highly evolved search engine libraries; Youssef has previously shown how an appropriate 'textualization' of mathematics into an indexable form allows standard text search engines to be applied.

Furthermore, given sufficiently semantic source forms for the math, such as LaTeX or Content MathML, the indexed form can be enhanced by co-locating synonyms, aliases and other metadata, thus increasing the accuracy and richness of expression.

Unfortunately, Content MathML is not always available, and the conversion from LaTeX to Presentation MathML (pMML) is too complex to carry out on the fly. Thus, one loses the ability to provide query-specific, fine-grained highlighting within the pMML displayed in search results to the user.

Where semantic information is available, however, such as for pMML generated from a richer representation, we propose augmenting the generated pMML with those semantics from which synonyms and other metadata can be reintroduced. Thus, in this paper, we aim to have both the high accuracy introduced by semantics while still obtaining fine-grained highlighting.

1 Introduction

The achievements of modern text search on the web have raised standards and user expectations. The relevance of the top ranked results to the query are often astounding. Concise summaries of the search results with matching terms highlighted allows users to quickly scan to find what they are looking for. Search has become, for better, sometimes worse, one of the first tools used to solve many information problems. These high expectations are carried over to math search; anecdotally, we see users uninterested in its unique challenges — the chess playing dog rationalization[1] carries little weight.

[1] He doesn't play well, but that he plays at all is impressive.

S. Autexier et al. (Eds.): AISC/Calculemus/MKM 2008, LNAI 5144, pp. 536–542, 2008.

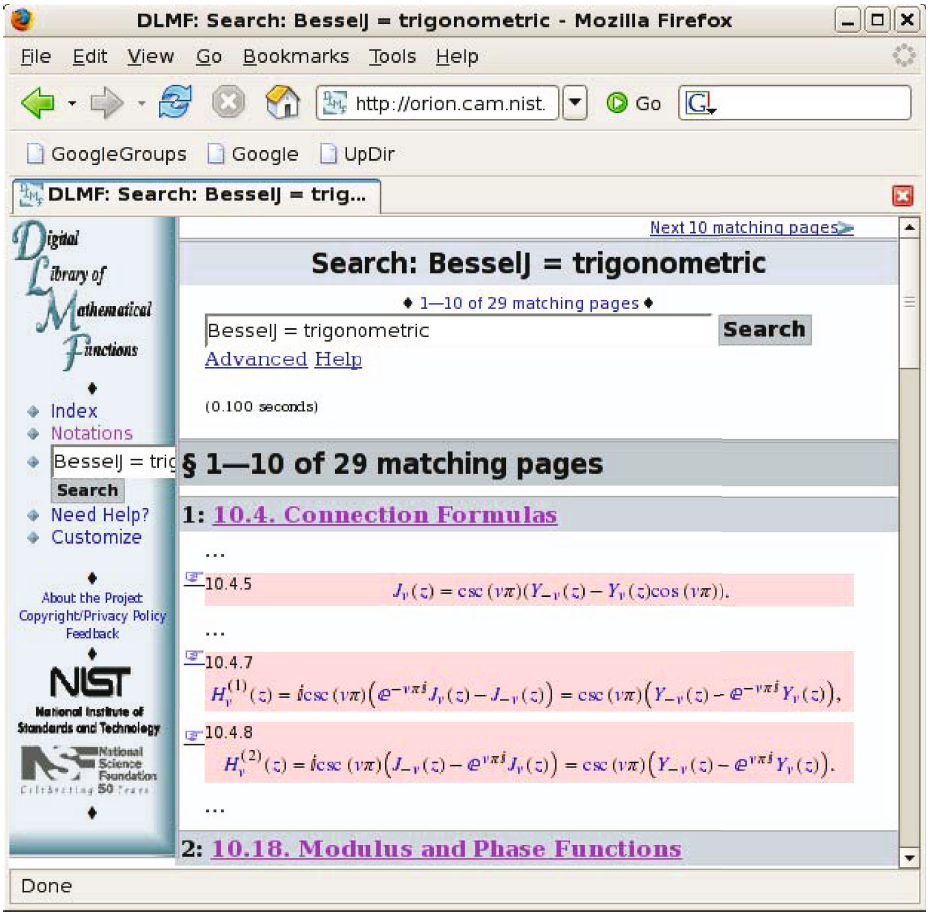

Fig. 1. Search results for `BesselJ = trigonometric` with coarse-grained math highlighting

In previous work [1], we described a strategy for math-aware search in which the mathematics, in whatever representation, is encoded into a linear textual form that can be processed by a conventional text search engine. This allows us to leverage the advances and tools in that field[2]. Recently, we have reported progress in improving the accuracy and ranking of math-aware search through the embedded semantics [4,5], and work in these areas continue.

In this paper, we describe on-going work in which we will add fine-grained highlighting to our math search engine. Our test corpus is the Digital Library of Mathematical Functions[2]. In the search results 'hit list', we show a summary of each matching document containing the fragments that match the query. Rather than merely highlighting complete math expressions that have matches

[2] http://dlmf.nist.gov/

(See Figure 1 for an example), we intend to refine the presentation so that only the individual terms within the MathML[3] that match are highlighted. The expectation is that this will help users scan the hit list to select the most appropriate ones. Of course, this must be accomplished without sacrificing the previous improvements.

2 The Role of Semantics

Document processing typically proceeds through several levels of transformation beginning with the 'authored' form possibly passing through semantically-enhanced forms (e.g. some XML variant), and ending up in a presentational form for display to the user (e.g. HTML). In our case, the authored form is LaTeX; we use LaTeXML [3] to carry out the conversion which involves TeX-like processing, expansion, parsing of the mathematics, and data reorganization, while preserving as much authored information (both presentational and semantic) as we can — the process is fairly time-consuming for the large documents involved, and is clearly better suited for batch processing than server-side usage.

As we will show, the semantic form is the most useful for many information extraction tasks, particularly indexing for search. The highlighting necessarily involves the presentation form. The cooperation between these levels for the purposes of search, is influenced by what information can be carried between levels, and whether the conversions are efficient enough to do on-the-fly, in the server. In particular, we must decide at exactly which level search indexing is applied.

Mathematical notation is extremely symbolic; much information can be conveyed by a single letter or slight change in position. A `J` may stand for the Bessel function J_ν, or the Anger function $\boldsymbol{J_\nu}$, or perhaps Jordan's function J_k. Of course, any content-oriented markup worth its salt will make these distinctions clear.

Yet, a user may search for a specific function, say `BesselJ`, expecting only matches to the first function, or may search more generically for `J` and reasonably expect to match all of the above functions. Thus, for good accuracy and expressiveness, the indexed form of math needs to reflect both semantic and presentational aspects. A technique we call token *co-location* allows storing multiple forms of a term in the search index. Aliases, synonyms and other metadata can also be handled in this way, such as associating 'trigonometric' with sin and cos. The query in Figure 1 demonstrates these aspects. Furthermore, by co-locating the aliases, they are stored as if they appeared at the same document location, preserving information which will be essential for highlighting.

It is important to note at this point that, given the smaller corpus of a digital library like the DLMF[1], such techniques as latent semantic indexing are harder to apply, but may be worth investigating in the future to turn latent semantics into explicitly stated semantics for richer search capabilities.

[3] http://www.w3.org/Math/

3 The Role of Presentation

If the application needs only to find which documents match a query, one has much flexibility regarding at which level of processing the document is indexed. However, it is very useful to a reader to see the portions of the matching documents that match the query, along with some context. Given that a search query is at best a crude approximation to what they really wanted, such summaries help them select the most documents from the results.

In the case of text documents, a common approach is to break each matching document into fragments (typically sentences) at search time. The query is then reapplied to the fragments to select the relevant ones. Finally, the matching tokens within each fragment are then marked for highlighting.

This approach needs to be refined to deal with richly structured XML documents. In such cases, reasonable summary fragments can include not only sentences, but equations, figures and portions of tables. Moreover, the reconstructed summary must itself also be a valid XML document. Furthermore, one would like to store the fragments as XML document fragments, and thus the indexing must 'skip over' the XML markup. In fact, this is easily achieved by manipulating term positions during indexing, and thus the eventual highlighting preserves the XML structures.

This additional complexity is a strong inducement to carry out the fragmenting in batch mode, rather than during search, and to generate a separate fragment index. At search time, the query is applied to the document index to find the relevant documents, as before. But, to summarize each document, the fragment index is searched to find matching fragments for that document.

Moreover, this argues for indexing the documents at the presentation level, in contrast to the arguments of the previous section. This greatly simplifies the processing needed at search time — one never gives up the hubris that one's site will be The Next Big Thing. However, token co-location again can be used to associate any desired semantics with the presentational tokens, provided they are made available.

Turning to the mathematics specific aspects of this argument, we note that in our previous work, each math entity was treated as a unit by converting it to a textual form corresponding to the LaTeX markup originally used to author it. Since we have adopted a semantically enhanced LaTeX markup (e.g. macros like `\BesselJ`)[3], we gain the benefits discussed in the previous section. However, converting from LaTeX form to presentation on-the-fly is not feasible, and so we lose the direct correspondence with the structure of the Presentation MathML. Consequently, we are unable to highlight individual mathematical terms or variables within the summary, but are only able to highlight the entire formula, as can be seen in Figure 1.

4 A Strategy

We propose the following strategy to work around the limitations and contradictions described in the previous sections. Firstly, we will convert from the

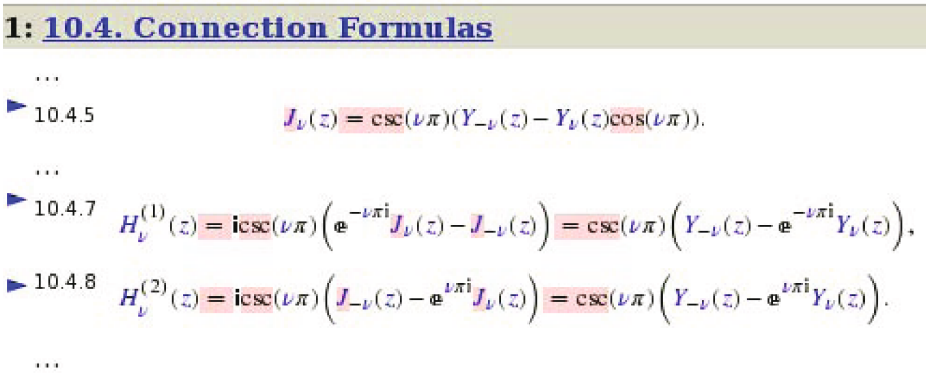

Fig. 2. The query of Figure 1 with fine-grained highlighting

content-oriented level of representation to the presentation-oriented level while augmenting the presentation with any semantic information available. In our particular case, we can simply carry the 'meaning' attribute of LaTeXML's internal representations over to the generated Presentation MathML token elements in a special attribute, say `ltxml:meaning`. These 'meanings' form a rough ontology, and could be (but have not yet been) defined by OpenMath[4] Content Dictionaries. Associated with each such meaning is a set of keywords and aliases. Thus, associated with the `sin`, generated by `\sin`, are the keywords: trigonometric, sine, function, elementary and periodic. A similar approach could be used when converting Content MathML or other representations.

A careful treewalk of the augmented MathML generates the same textualization as was produced by textualizing the LaTeX, as described above. In this case, however, we take care to manipulate the positions of the textualized tokens as they are being indexed so that each corresponds to the appropriate MathML fragment with the expression. This assures that the eventual highlighting will be both possible, and that the result will be well-formed XML. Whenever a token with assigned meaning is encountered, the additional aliases are co-located with the principal MathML token.

5 Preliminary Results

A preliminary implementation of these ideas has been carried out, with the result as shown in Figure 2. Although no detailed benchmarking has been carried out, there appears to be no execution time penalty for either indexing or search.

Two kinds of highlighting inaccuracies could be expected from this relatively simple approach.

[4] http://www.openmath.org/

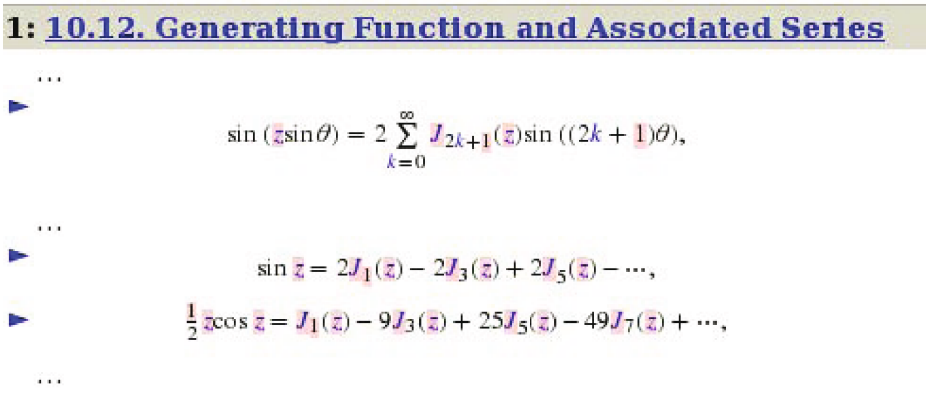

Fig. 3. First hit from the query `J_1(z)`

– tokens present in the query but not matching the full query may be highlighted when they should not;
– non-indexed tokens or markings (e.g. parentheses, surds) may fail to be highlighted when they should be.

And in fact, the first is seen in Figure 2 and both are demonstrated in Figure 3.

Although subjectively exploring the DLMF corpus with fine-grained highlighting is more pleasant and helpful than the coarse-grained whole-formula highlighting, the errors, especially the first type can be quite distracting; the second problem is more an aesthetic issue.

In fact, the over-highlighting problem is commonly seen in text search engines, but it simply is not as disturbing as within math. For example, in Figure 3, every occurrence of z is highlighted, not just when it is the argument of J_1. Users may find this confusing, and it begins to defeat the purpose of the fine-grained highlighting, namely to make easy the selection of good matches.

The solution will be to apply more of the restrictive 'logic' of the query (ands, ors, proximity) to the highlighting phase than is currently done. The search engine we are using, Lucene[5], apparently only makes token positions available when searching using primitive token queries. If that is indeed the case, we will be forced to reimplement or approximate some of that logic during highlighting. Tedious, but doable.

The second problem seems less serious but may yield to one of two approaches. One would be to extend the textualization language to allow indexing MathML structural schema rather than just token elements. Another would be to broaden the highlighting to a parent when many children of a node are highlighted.

[5] http://lucene.apache.org/

6 Conclusion

We have described techniques that should give math-aware search engines the capabilities of both high accuracy and relevance, while still presenting the user with a concise and appropriate summary of each result that indicates how the document relates to the query.

This paper describes work that is in progress. However, initial implementation gives satisfying results, with some flaws. We anticipate being able to remedy some or all of those flaws and demonstrate the results at the conference.

References

1. Miller, B., Youssef, A.: Technical Aspects of the Digital Library of Mathematical Functions. Annals of Mathematics and Artificial Intelligence 38, 121–136 (2003)
2. Baeza-Yates, R.A., Riberto-Neta, B.(eds.): Modern Information Retrieval. Addison Wesley, Reading (1999)
3. Miller, B.: Creating Webs of Math Using LaTeX. In: Proceedings of the 6th International Congress on Industrial and Applied Mathematics, Zürich, Switzerland, July 17 (2007)
4. Youssef, A.: Information Search And Retrieval of Mathematical Contents: Issues And Methods. In: The ISCA 14th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE-2005), Toronto, Canada, July 20-22 (2005)
5. Youssef, A.: Methods of Relevance Ranking and Hit-content Generation in Math Search. In: The 6th Mathematical Knowledge Management Conference, Hagenberg, Austria, June 27-30, pp. 393–406 (2007)

Disclaimer: Certain products, commercial or otherwise, are mentioned for informational purposes only, and do not imply recommendation or endorsement by NIST or GWU.