Math Aware Search

Mar 27, 2018

CONTENTS

CHAPTER

ONE

ABOUT YOU

- Name: Guruprasad Hegde
- E-mail address: guruhegde1308@gmail.com
- IRC nickname(s): ghegde
- Any personal websites, blogs, social media, etc: None, will create one to post project updates
- github URL: https://github.com/guruhegde
- Biography:

I am a master student in computer science at the University of Saarland, Germany. I am from Karnataka, India.

I choose courses which has project work as part of it, this helped me improve my overall programming skills(coding,testing, debugging). I am hands-on with tools like gdb, valgrind.

During my spare time, I learn about new programming languages or read code from github repos and pick up the design, coding styles etc.

1.1 Background Information

Have you taken part in GSoC and/or GCI (https://codein.withgoogle.com/) and/or similar programmes before? If so, tell us about how it went, and any areas you would have liked more help with.

I don't have any prior experience with GSoC or GCI or similar programmes.

Please tell us about any previous experience you have with Xapian, or other systems for indexed text search.

I have been playing with xapian and some of its tools for the past couple of months. Apart from user experience with google, duckduckgo for long time, I don't have any other experience with search engine.

Do you have previous experience with Free Software and Open Source other than Xapian?

I contributed patches to Shogun(ML library), CLTK. PR links:

- https://github.com/shogun-toolbox/shogun/pull/4144
- https://github.com/shogun-toolbox/shogun/pull/4119
- https://github.com/shogun-toolbox/shogun/pull/4116
- https://github.com/shogun-toolbox/shogun/pull/4139
- https://github.com/shogun-toolbox/shogun/pull/4123
- https://github.com/cltk/cltk/pull/660
- https://github.com/cltk/cltk/pull/661

What other relevant prior experience do you have (courses taken at college, hobbies, holiday jobs, etc)?

Courses i have taken which are relevant:

- Statistical Natural Language Processing
- Information Retrieval & Data Mining
- Database Systems
- Distributed Systems

What development platforms, tools and methods do you prefer to use?

OS - Arch Linux, editor - vim, version control - git

Have you previously worked on a project of a similar scope? If so, tell us about it.

I would like to mention couple of projects(academic), which highlight the necessary background and experience.

1.In-Memory Database System(c++/13kLoC): I implemented the following database system components.

- Indexing techniques : Hash table, B+ Tree.
- data layout : row store, column store.
- Compression Techniques: RLE, Dictionary compression.

With this project, I had hands-on experience with building system. The project used latest c++ features(c++14, c++17), helped me get hands-on with them.

2.Yet Another File System(c++/7kLoC): I implemented a multi-server file system.

- File system with operations - mkdir, read, write, remove
- implement lock server, extent(data) server and support caching.

I used low level system apis(like system calls), fuse library and RPC library.

What timezone will you be in during the coding period?

CET(Central European Time)

Will your Summer of Code project be the main focus of your time during the program?

Yes. I signed up for a seminar at the university. This requires 2 hours/week meeting time and upto 8 hours/every 2 week for project work. This can be adjusted with some time during the weekend also if necessity arises. Other than this I don't have any commitments like part-time job, course work etc. I want to take GSoC as full time job as required.

Expected work hours (e.g. Monday–Friday 9am–5pm UTC)

Monday-Friday 8am-4pm UTC+1, half day off during the week to attend seminar.

Are you applying for other projects in GSoC 2018? If so, with which organisation(s)?

No.

CHAPTER

TWO

YOUR PROJECT

2.1 Motivations

Why have you chosen this particular project?

- I remember myself searching for Math notation and some strange symbol meaning in google, then results were not really satisfactory. I feel 'math aware search' is a useful feature to be part of search engine. Hence this topic drew my attention more than other topics.
- Recently, I had heard about the research in temporal IR which was very interesting. I found math aware IR eqully interesting.
- Having gone through the literature, I found that some researches are very recent and it's an emerging field. This motivated me to take up this topic.

Who will benefit from your project and in what ways?

Currently search engine support for retrieving math formula query is not very good, as it treats them as text data, resulting in poor performance with search hits. With search engine support for math queries, one will be able to get a better search result when searching for math formula of interest. This is useful for students, people in academics. A researcher can find applicability of particular math equation in research papers by giving equation as query. Currently he/she has to put the math formula name or search for concept name.

2.2 Project Details

Describe any existing work and concepts on which your project is based.

The system I propose is based on Tangent - math expression search engine created by David Stalnaker and it's improved versions developed at dprl, RIT[4]. Tangent performs indexing on the structure of the expression represented in MathML format. Encoding of structure is done by constructing symbol layout tree from the expression and then generating symbol pairs from that tree. The symbol pairs are indexed using inverted index.

Do you have any preliminary findings or results which suggest that your approach is possible and likely to succeed?

Tangent system code is available for public. I successfully ran the system and reproduced some of the results. Also, team who developed Tangent participated in NTCIR math retrieval task[5] and obtained competitive results for arXiv-main task[1].

What other approaches to have your considered, and why did you reject those in favour of your chosen approach?

[4] https://www.cs.rit.edu/~dprl/Software.html#tangent-s
[5] http://ntcir-math.nii.ac.jp/
[1] Tangent - http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings12/pdf/ntcir/OVERVIEW/01-NTCIR12-OV-MathIR-ZanibbiR.pdf

I considered MIaS system[2], which linearizes math expression into text and uses traditional text retrieval search engine. Indexing part is similar to the Tangent system, approach I would like to propose. It differs in the preprocessing of math expression to generate indexes. MIaS uses tokenization of expression and performs unification[7] strategy to generalize the expression. The reason for selecting the approach used in Tangent system is, symbol layout tree approach extracts the structural information in expression better than tokenization of text. This is corroborated by the findings of hypothesis test. Also, Tangent system performance is better than MIaS system.

I also studied the system[3] implemented by MCAT group. They include textual context of the formulae and integrate retrieval of text and formulae. The system extracts three granularity level text information. Even though this system performs better than Tangent, due to the complexity and keeping time constraint in mind, I planned to pick Tangent system. This could be tried in Xapian in future.

Please note any uncertainties or aspects which depend on further research or investigation.

> We have decided to support latex format for documents and queries. This requires conversion of latex format to mathml format. As this function is in itself can be a project, we plan to use third party tool like latexmathml. But the exact tool choice is not finalised yet. This feature will be floater in the work pipeline.

How useful will your results be when not everything works out exactly as planned?

Most of the work is divided into small independent chunk in timeline. So there will be useful result available after completion of each block. Hence extending the work in future will be very easy.

2.3 Project Timeline

Project Structure

Indexing stage:

1. Extract List of Math expression from Document - this involves searching the document and extract MathML element. If document is in latex format, conversion of math expression from latex to mathml is performed.
2. Generate symbol layout tree for each math expression.
3. Create a list of symbol pair tuple from the symbol layout tree in step 2.
4. Symbol pair tuple is stringified. The output string is similar to the term in case of text document. I will call this math term.
5. Math term from step 4 is indexed using Xapian's existing indexing system.
6. Steps 2 - 5 are repeated for each math expression in a document.

Searching stage:

1. Convert Query to Symbol Layout tree structure.
2. Generate symbol pair tuple from symbol layout tree.
3. Formulate disjunction query from symbol pair tuples.
4. Retrieve relevant documents based on dice similarity coefficinet metric.

Details on the symbol layout tree structure and symbol pair tuple:

Symbol layout tree is used to represent each math expression. The vertices in this tree represent the symbols in the expression and the edges are the spacial relationship between symbols. The tree is rooted at the leftmost symbol.

[2] NTCIR12 - http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings12/pdf/ntcir/MathIR/05-NTCIR12-MathIR-RuzickaM.pdf
[7] All variables and constants are substituted with unified symbols.
[3] MCAT - https://pdfs.semanticscholar.org/6995/bf023d37a5fc10fe60d3783772801994751d.pdf

Symbol pair tuple of the form (s1, s2, R, #) with ancestor symbol s1, descendant symbol s2, edge label R from s1 to s2 and a count(#) is generated by traversing from the root of symbol layout tree. Parameter window size(w) control the maximum path length between symbols in tuples and other parameter to decide whether to include tuples for symbols at the end of writing lines(EOL).

Pseudocode for indexing, searching, ranking:

```
Index(expression, index):
for pair in symbol pairs of expression:
    append expression to index[pair]

Search(query, index):
for pair in symbol pairs of query:
    for expession in index[pair]:
        append pair to result_pair[expression]
    sort expressions by the ranking function(using result_pairs)
    return expressions
```

Timeline:

Until May 10 :

- Get to know the community, interact with the people.
- Read and understand the Xapian code base -understand the underlying design principle, get to know all the relevant classes.
- Submit patches for existing issues, go through code review process.
- Get equipped with all the background knowledge needed to implement the project parts - writing parser, adding weighting scheme, study how wildcard expansion performed.
- Have clear blueprint of the project.

1. Preprocessing stage

[block 1: May 14 - 15]: Implementation to extract list of presentation mathml expression from the input document.

[block 1: May 16 - 21]: Write symbol layout tree class. Add the necesaary attributes, implement member functions. Task requires representing math symbols as different types of nodes and spatial relationship as edge types, writing helper functions to traverse the tree, adding children, updating the tree etc.

[block 1: May 22 - 28]: Construct symbol layout tree from presentation mathml expression. This involves parsing the mathml expression and adding the extracted token to the tree structure.

[block 1: May 29]: Make sure test cases are there for the code written so far. Write documentation.

[block 1: May 30]: Buffer to cover up any lagging work if any.

[block 1: May 31 - June 1]: Create symbol pair tuple class, make symbol pair tuple class indexable.

[block 1: June 2 - 4]: Generate symbol pair tuple from symbol layout tree with given window size parameter.

[block 1: June 5 - 6]: Make sure test cases are there for the code written so far. Write documentation.

[block 1: June 7 - 11]: Integrate the work done so far. Rework the class design, refactoring the code if needed.

Deliverable:

Given a document containing math ml expression, set of symbol pair tuple generated.

2. Indexing stage

[block 2: June 18 - 19]: Work on indexing math terms available at the end of block 1. Implement posting list for math terms.

[block 2: June 20 - 21]: Test indexing of documents with multiple test data files. Fix issues if any.

3. Searching stage

[block 2: June 22 - 25]: Implement dice's coefficinet of similariy weight metric.

[block 2: June 26 - 28]: Test the weight metric with multiple test data. Fix issues if any. Add documentation.

[block 2: June 29]: Buffer time. Work on anything lagging, else take a long break.

[block 2: July 2 - 4]: Construct symbol layout tree from the query input. This involves majority code reuse from block 1. Handle query specific changes needed.

[block 2: July 5 - 9]: Implement document retrieval from the given query. This involves generating symbol pair tuples from the symbol layout tree for the query and fetching postings from the database index.

Deliverable:

Indexing of documents containing math expression. Retrive documents for the given math query.

[block 3: July 12 - 15]: Implement latex to mathml converter for document and query containg latex math expression. Write test and document it.

[block 3: July 16 - 20]: Integrate the code and perform testing. Code refactor if needed. Document the code. Profile the code, evaluate the performance.

[block 3: July 23 - 24]: Do the house keeping work in this period. Address any pending requested changes and fix issues etc.

Deliverable: Basic math aware search functionality will be realised in xapian.

Add-ons

[block 3: July 25 - 27]: Implement support for wildcard queries. This requires extending tuple generation module of query further.

[block 3: July 30 - August 2]: Add support for math expressions containg matrix type.

[block 3: August 3 - 6]: Matrix support requires update to parser code and tuple generation code. Write tests and make sure no breakage in any functionality.

[block 3: August 7 - 14]: Complete any review modifications pending. Finalize the documentation. Do clean up work if any present.

Stretch Goal: Adding re-ranking stage - as this process is quite complex, it is attempted only if at least a week time remains after the completion of promised deliverables.

2.4 Previous Discussion of your Project

Project Discussion on mailing list thread link[6]

2.5 Licensing of your contributions to Xapian

Do you agree to dual-license all your contributions to Xapian under the GNU GPL version 2 and all later versions, and the MIT/X licence?

For the avoidance of doubt this includes all contributions to our wiki, mailing lists and documentation, including anything you write in your project's wiki pages.

[6] https://lists.xapian.org/pipermail/xapian-devel/2018-March/003243.html

Yes. I agree.

2.6 Use of Existing Code

If you already know about existing code you plan to incorporate or libraries you plan to use, please give details.

I plan to use third party library to convert latex to mathml. Library is not decided yet.