Integrating WordNet into Xapian

April 09, 2019

1 About You **2**

1.1 Background Information 2

2 Integrating WordNet into Xapian **4**

2.1 WordNet, a lexical database being used to provide ontological input to analyse relevance of documents while improving the detection of query intent. 4

2.2 Project Details 4

2.3 Project Timeline 6

2.4 Previous Discussion of your Project 7

2.5 Licensing of your contributions to Xapian 7

2.6 Use of Existing Code 7

CHAPTER

ONE

ABOUT YOU

- Name: Tejasvi S Tomar
- E-mail address: tstomar[at]outlook.com
- IRC nickname(s): tstomar
- Any personal websites, blogs, social media, etc: http://t.me/tstomar (telegram) http://tejasvi.github.io (stem blog)
- github URL: http://github.com/tejasvi
- Biography:

  I am enthusiastic about technology and its impact on people lives. Currently I'm pursuing bioengineering as an undergrad. The field of computational biology, genomics, and neuroscience interest me the most. I've been fascinated with computers and biology since I was a kid. And ever since wanted to work in this interdisciplinary field. Of late, I'm also intrigued with AI, augmented reality and philosophy behind them. I look forward to contribute in making of the exciting future ahead.

  When I'm off screen, I have many and varied interests. Currently I swim and play water sports. I'm an avid book reader and occasionally write. And if there is time at hand I engage in philosophy too.

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 wish I were aware of GCI earlier. However this is my first chance to apply for GSoC.

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

Nothing major projects I worked on before involved search systems. Though I had conceptual idea of workings of search engine which is more concrete now after getting involved with Xapian.

Tell us about any previous experience with Free Software and Open Source other than Xapian.

I came accross linux as my eight years old PC started slowing down. Since then I have tried tens of distros and various utilities present within. Though my knowlege expanded more or less within the scope of general computing and only recently the scope has broadened to development. And I'm loving it so far and excited to be the part of the community.

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

Relevant courses taken in college are general Mathematics (calculus, probability, matrix algebra, real analysis,...) and computer science (data structure and algorithms, database management,...). Besides, I'm fairly skilled in navigating Internet an upshot of often breaking setup more than often.

Few vaguly relevant ones; I have learned web development a bit being part of the team for our college fest website. Besides I have worked with microcontrollers as part of robotics club. I also study mathematics specific to ML lately.

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

Currently I use vscode as the editor. On the OS side I've tried plethora of configurations and found Ubuntu with KDE to suit me well. However I'm not reluctant to try out new tools if the situation demands.

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

Though none matching same scope, I have completed a moderately sized robotics project. The goal was to balance balls on movable plate using microcontroller. The code involved was in C++ and algorithm for balancing was derived from a research article. The idea of implementation was to get coordinates of ball and predict the tilt of the plate based upon current, past, and target coordinates of the ball.

Project repo: https://github.com/tejasvi/ballet-of-bots

What timezone will you be in during the coding period?

UTC +5.5

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

Yeah, it will be for sure.

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

I will spend at least six hours a day. Also I would prefer to spread out my work equally over week days. It translates to at least 42 hours of work for a week. Besides, I'll have extra 3-4 hours to commit during summer break or on holidays.

Are you applying for other projects in GSoC 2019? If so, with which organisation(s)?

Since here I found skillset required to be most holistic, I would prefer to work exclusively with Xapian for now.

CHAPTER

TWO

INTEGRATING WORDNET INTO XAPIAN

WordNet, a lexical database being used to provide ontological input to analyse relevance of documents while improving the detection of query intent.

Why have you chosen this particular project?

For one, this project involves application of the academic research, something I relish. Besides, integration of WordNet in Xapian has potential to extend it beyond conventional use cases and improve upon existing ones. A main problem with the current search engines is the large volume of documents extracted as a result of broad, general queries, and the lack of output produced to specfi c, narrow questions. Current method for matching terms directly with document index not takes the sense diversity of words into account. I am improving Xapian in this area.

Who will benefit from your project and in what ways?

Out of seven billion people only five percent of them speak english natively. Moreover, there are only one billion total english speakers while more than 50 percent web content is in english language. This creates a language barrier for rest six billions to access majority of information as their vocabulary is not evolved enough to frame their query efffectively.

WordNet integration will facilitate such people by adding more flexibility to interpretation of certain queries in particular.

Besides above, the most benefitted usecases:

- Search pool containing highly heterogenous information or,
- The user is not well familier with the resource jargon

These constitute majority of the userbase. This project will work to quantify intent of queries and judge relevance of a document by improving upon existing term match methodology using WordNet semantics data.

Source: https://en.wikipedia.org/wiki/Global_Internet_usage

Project Details

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

Plenty of papers have been published to demonstrate the potential of semantic databases to improve text based search considerably. WordNet is a well developed and widely used such lexicon which is to be integrated into Xapian. There are multiple fronts possible to add the integration. Currently, query expansion, incremental search, weighting scheme, and an inbuilt synonymn dictionary will be most rewarding areas to start with.

The project is sub-divided such that to ensure modularity at small levels. Broadly speaking, first goal will be to create a handy interface to WordNet. Since full integration will extend beyond the project timeline, the base class demands high extensibility. Therefore future integration prospects are considered while prototyping. After the interface is developed and tested, each incorporation opportunities can be targeted one by one.

Custom index will be used to fetch lexical data. It is done to minimize overhead due to the information not required by features. Database formats currently planned to support for indexing are WordNet, Open Multilingual Wordnet (tab seperated), and Lexical Markup Framework.

Deliverables: 1. A framework to interface WordNet with Xapian 2. Built-in *synoynmns* dictionary as a result 3. Revamped query expansion using WordNet specific algorithms 4. Enabling incremental search using query expansion and NLP techniques

Strech goals: * Enhancing query expansion by displaying alternative intents * Incorporating WordNet into LETOR and weighting schemes

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

The advantages of incorporating lexical information are evident from the fact that currently most web search engines uses some form of natural language processing. As the searchable data increases, topics become more important than keywords. Context aware searches require lexicon to *tokenize** topics. A 2012 research demonstrated the advantages of contextual searches both quantitatively and qualitatively over keyword based.

> Children's web search with Google: the effectiveness of natural language queries[1]

Additionaly abundant research has been done to exploit the WordNet data to improve *recall* and *precision* concurrently though they affect each other inversly in general. Following are few representative papers attesting to it.

> Using WordNet and Lexical Operators to Improve Internet Searches (paywall)[2]
>
> Query expansion via wordnet for effective code search[3]
>
> The informative role of WordNet in open-domain question answering[4]

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

Limitation which is being addressed is the lack of intent detection for queries which are imprecise or have multiple interpretations. One of the approach could be the use of neural networks to build user intent model. However conventional uses of Xapian will present problem of cold start or can't aggregate query data quickly enough for a funtional model.

For accessing lexical data direct api calls to WordNet or equivalent can also be made. However they are likely to cause overhead compared to self-desigened index which can be optimised for the implemented features.

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

There are no apparant prospects of significant decrease in search efficiency in terms of speed, considering research papers, when lexical databases are employed. However it will still be something to warrrant caution while implementing related features. Moreover the algorithms which look good on paper could involve skewed sample for testing. Besides they may not perform well in Xapian specific environment. Such setbacks certainly have possibility to digress the planned project outline.

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

Since the project is divided into individual modules *everything* not working out well would only lead change in feature priorities. Moreover in an implausible scenario, WordNet integration will be done in smaller extent than initially planned. In that case other than already incorporated benefits, we will have WordNet interface as a firm bedrock to build upon.

[1] https://dl.acm.org/citation.cfm?id=2307121

[2] https://dl.acm.org/citation.cfm?id=613476

[3] https://ieeexplore.ieee.org/iel7/7066219/7081802/07081874.pdf

[4] https://dingo.sbs.arizona.edu/~sandiway/csc620/eggers.pdf

Project Timeline

Preceeding May 27

- Enhancing grasp of Xabian codebase pertaining to the project goal
- Start prototyping classes (and discussing) to implement later
- Keep researching for better algorithms for feature implementations
- Understand basic ML concepts for possible LETOR integration
- Wrapped up in month of April for exams

The summer of code starts!

Week 1: June 3

- Implement base class WordNet
- Implement WnIndexer to create custom index from various inputs Intially index format will be basic to be augmented later as required.

Week 2: June 10

- Test WnIndexer (and WordNet) extensively and document
- Implement GetSets to get *synonymns* from desired synsets which could also contain additional information obtained after further processing. E.g. relation of synsets across query terms

Week 3: June 17

- After implementation of basic functionality of GetSets like returning unproccessed synsets, implement more complex standard operations on synsets.
- Test GetSets and WnIndex individually then in synergy. Document the implementation.

Week 4: June 24

- Create WQueryParser for a seperate parser for wordnet
- Implement basic functionality deriving from QueryParser
- Test it individually then with previous implementations.

Phase 1 evaluation

Week 5: Jul 1

- Start implementing few WordNet specific operations in WQueryParser like injecting expanded terms corresponding to sentence structure
- Test the implementation with previous work

Week 6: Jul 8

- Implement query expansion algorithm in WQueryParser
- Test it against representative data
- Repeat above steps until results are satisfactory.

Week 7: Jul 15

- Test the whole implementation together
- Add support for built-in synonymns support by linking WordNet index to QueryParser. Also test it.

Week 8: Jul 22

- Integrate WQueryParser with QueryParser

- Test the implementation as this will mark first successful WordNet integration into Xapian. Document WQueryParser.

Phase 2 evaluation **Week 9: Jul 29**

- Create IncSearch class for incremental search implementation
- Implement basic operations such as additional string output.

Week 10: Aug 5

- Test the class against the build.
- Start implementing advanced standard operations required for algorithm.

Week 11: Aug 12

- Test the implementation again.
- Implement incremental search algorithm and test it against sample input

Week 12: Aug 19

- Do complete rigorous testing of all features added.
- Buffer zone for documentation and pending work

Week 13: Aug 26

- Buffer period for merging
- Fix bugs if any
- Start stretch goals

Phase 3 Evaluation

Previous Discussion of your Project

I've discussed it exclusively on IRC.

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.

Yes, I'm glad to do so.

Use of Existing Code

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

- **'WordNet <https://wordnet.princeton.edu/'_**
- **'Freeling <nlp.lsi.upc.edu/freeling/index.php/node/1'_**
- **'wordnet-blast <https://github.com/jardon-u/wordnet-blast'_**