Suggesting Query Completions for Xapian

Pallavi Gudipati
Tanmay Dhote
Siddhant Mutha
Parth Joshi
Mohammed Shamil

Department of Computer Science and Engineering

Outline

1. Introduction
    - Xapian
    - Query Auto-completion
    - Query Expansion
    - Motivation
2. Design aspects and Modifications
    - Logs and Trie
    - Query Expansion
3. Concepts from theory
    - Flyweight pattern
    - Decorator pattern
    - Factory pattern
4. Learnings
5. Difficulties

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Xapian

- Xapian is an Open Source Search Engine Library, written in C++.

Xapian

- Xapian is an Open Source Search Engine Library, written in C++.
- Highly adaptable toolkit which allows developers to easily add advanced indexing and search facilities to their own applications.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Query Auto-completion

- Autocomplete involves the program predicting a word or phrase that the user wants to type in without the user actually typing it in completely.

Query Auto-completion

- Autocomplete involves the program predicting a word or phrase that the user wants to type in without the user actually typing it in completely.
- The list of query candidates is generated according to the prefix entered by the user in the search box and is updated on each new key stroke.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Query Expansion

- Process of reformulating a seed query to improve retrieval performance in information retrieval operations.

Query Expansion

- Process of reformulating a seed query to improve retrieval performance in information retrieval operations.
- Query expansion involves techniques such as:

Query Expansion

- Process of reformulating a seed query to improve retrieval performance in information retrieval operations.
- Query expansion involves techniques such as:
    - Finding synonyms of words, and searching for the synonyms as well.

Query Expansion

- Process of reformulating a seed query to improve retrieval performance in information retrieval operations.
- Query expansion involves techniques such as:
    - Finding synonyms of words, and searching for the synonyms as well.
    - Finding all the various morphological forms of words by stemming each word in the search query.

Query Expansion

- Process of reformulating a seed query to improve retrieval performance in information retrieval operations.
- Query expansion involves techniques such as:
  - Finding synonyms of words, and searching for the synonyms as well.
  - Finding all the various morphological forms of words by stemming each word in the search query.
  - Fixing spelling errors and automatically searching for the corrected form or suggesting it in the results.

Query Expansion

- Process of reformulating a seed query to improve retrieval performance in information retrieval operations.
- Query expansion involves techniques such as:
  - Finding synonyms of words, and searching for the synonyms as well.
  - Finding all the various morphological forms of words by stemming each word in the search query.
  - Fixing spelling errors and automatically searching for the corrected form or suggesting it in the results.
  - Re-weighting the terms in the original query.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Motivation

- Xapian currently does not have any auto-completion or query expansion support.

Motivation

- Xapian currently does not have any auto-completion or query expansion support.
- To decrease the average time taken by a user to obtain a relevant search result.

Motivation

- Xapian currently does not have any auto-completion or query expansion support.
- To decrease the average time taken by a user to obtain a relevant search result.
- Query expansion gives us results that may be relevant to the user.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Logs and Trie

- Modified the *Xapian::Database* class to log the queries.

Logs and Trie

- Modified the *Xapian::Database* class to log the queries.
- Implemented *Xapian::Trie* classs for prefix matching.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

WordNet

- Wordnet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

WordNet

- Wordnet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
- Integrated WordNet for thesaurus support.

Synonym based Query Expansion

- Stemming - Process for reducing inflected words to their stem, base or root form, generally a written word form. Used *Xapian::QueryParser* class.

Synonym based Query Expansion

- Stemming - Process for reducing inflected words to their stem, base or root form, generally a written word form. Used *Xapian::QueryParser* class.
- Stop-word removal - Remove words from a list of common English stop words. Extended *Xapian::SimpleStopper* to *Xapian::PopualatedSimpleStopper* class.

Synonym based Query Expansion

- Stemming - Process for reducing inflected words to their stem, base or root form, generally a written word form. Used *Xapian::QueryParser* class.
- Stop-word removal - Remove words from a list of common English stop words. Extended *Xapian::SimpleStopper* to *Xapian::PopualatedSimpleStopper* class.
- Synonym Expansion - Expanding on Nouns. Implemented *Xapian::SynonymExpand* class.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Flyweight pattern

- The whole trie data structure in our code is represented using a light weight pointer to its root node rather than all nodes of the tree.

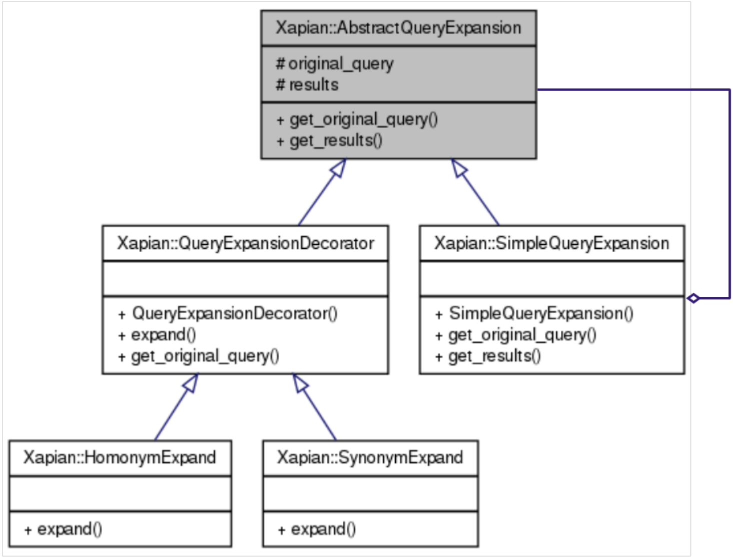
Flyweight pattern

- The whole trie data structure in our code is represented using a light weight pointer to its root node rather than all nodes of the tree.
- The data member root is a pointer of type *struct trie_node* and points to the root of the trie tree.

Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

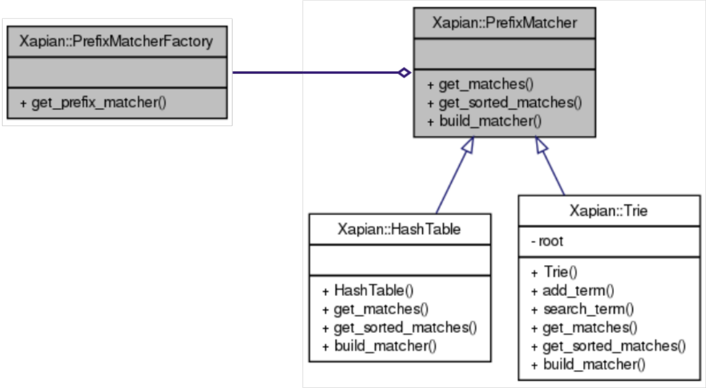
Decorator pattern



Outline

1. Introduction
   - Xapian
   - Query Auto-completion
   - Query Expansion
   - Motivation
2. Design aspects and Modifications
   - Logs and Trie
   - Query Expansion
3. Concepts from theory
   - Flyweight pattern
   - Decorator pattern
   - Factory pattern
4. Learnings
5. Difficulties

Factory pattern



Learnings

- Experience of working with an existing large code base of an Open Source Project.

Learnings

- Experience of working with an existing large code base of an Open Source Project.
- To inspect the library from a users point of view, we adopted the role of a user.

Learnings

- Experience of working with an existing large code base of an Open Source Project.
- To inspect the library from a users point of view, we adopted the role of a user.
- Interaction with the Open Source developer community.

Learnings

- Experience of working with an existing large code base of an Open Source Project.
- To inspect the library from a users point of view, we adopted the role of a user.
- Interaction with the Open Source developer community.
- How to integrate two different projects.

Learnings

- Experience of working with an existing large code base of an Open Source Project.
- To inspect the library from a users point of view, we adopted the role of a user.
- Interaction with the Open Source developer community.
- How to integrate two different projects.
- Tools like Doxygen, QtCreator etc.

Difficulties

- Finding the optimum position to insert new code due to the large existing codebase.

Difficulties

- Finding the optimum position to insert new code due to the large existing codebase.
- Integrating two existing large-scale projects - WordNet and Xapian.

Difficulties

- Finding the optimum position to insert new code due to the large existing codebase.
- Integrating two existing large-scale projects - WordNet and Xapian.
- Using the library we built from the point of view of the user.

References I

- Xapian
  Open Source Search Engine Library, http://xapian.org/.

- WordNet
  Lexical database of English,
  http://wordnet.princeton.edu/.

- Qt Project
  Cross-platform application and UI framework,
  http://qt-project.org/.

- Doxygen
  Tool for generating documentation,
  http://www.stack.nl/~dimitri/doxygen/.