Developing a Music Search Engine

Rashmi HTI, Sherin Thomas, Vinod Chhapariya

October 2016

1 Introduction

Musical Search Engine is web based project for searching the songs lyrics from different movies and albums. It is implemented using Apache Solr search platform. Apache Solr is Open source enterprise search platform written in Java. It is powered by Lucene, which is a Java liberary.

The main features of Solr are Full text search, Hit highlighting, Faceted search, Real-time indexing etc. It uses Inverted Indexing for the searching process. Web crawling is used to generate the data from set of Seed Url's.

This Project aims at the development of optimized search engine for the songs of different languages.

2 Work done - Phase 1

2.1 Systems Requirements

Java 8, Eclipse Java EE IDE, Solr 4.10.1, Jetty plugin,Google youtube service API.

2.2 Dataset generation

The dataset for populating the search engine is generated by crawling the seed url 'http://www.hindilyrics.net/lyrics/' .A recursive method of crawling is used to extract each of the unique lyrics pages starting from the given seed url.The following fields are extracted from each webpage.

- Page title
- url
- Movie
- Song title
- Singer(s)
- Music by
- Lyricist(s)
- Lyrics
- Metadata

The crawled data is stored into SolrDocument with the field values matching those specified in the schema.xml file. Each such document corresponding to a lyrics page is then added to the server. Jsoup libraries are used for crawling the pages.

2.3 Indexing and schema generation

• The field values on which the user can query are indexed by solr implicitly. The addition of documents in the data generation phase invokes POSTing commands to Solr to add documents. The fields are then indexed.

• A new Solr Core called Music is created for serving the purpose.

• The representation of each of the query fields are specified in the schema.xml file. The document id is declared as unique key.

• The data is tokenized, converted into lower case, and word stems are removed before indexing.

• Index time boosting is used to increase the relevance of documents based on particular fields.

• Solrj libraries are used for populating the solr server with the crawled dataset.

2.4 UI and search

• Request handlers are used for performing searches and returning the results.

• Search UI based on VelocityResponseWriter is used along with several features.

• Faceted searching is implemented on various fields.

• Search result grouping is implemented on some of the field values.

3 Work done - Phase 2

The following fields are extracted from each web page for implementing the features described below.

- Top Movie name
- Top Movie thumbnail image
- Top Movie url
- Movie thumbnail image url
- Singer profile url
- Youtube link

3.1 Movie thumbnails

• Thumbnail images for each of the movies that appears in the search results are included.

• The image urls are extracted and stored to serve this purpose.

3.2 Youtube links

• The links to the youtube videos for each of the song in the search result is added using Maven api.

• The song names are searched in youtube and the first result url is extracted and stored in Solr.

• The hyperlinks for the youtube video is provided in the search results and the video is dynamically fetched.

3.3 Latest releases

• The latest released movie lists along with the movie thumbnails are added to the homepage of the search ui.

3.4 Singer profiles

• The names of each singers of the songs are displayed as a hyperlink which can redirect to the corresponding profile pages for the singers.

3.5 Suggestions and spell checking

• It provides users with automatic suggestions of movie names while searching.

• It also provides inline query suggestions based on other, similar, terms in case of spell checks.

• The basis for these suggestions and spell check are the terms in a field that are stored in Solr.

4 Methodology

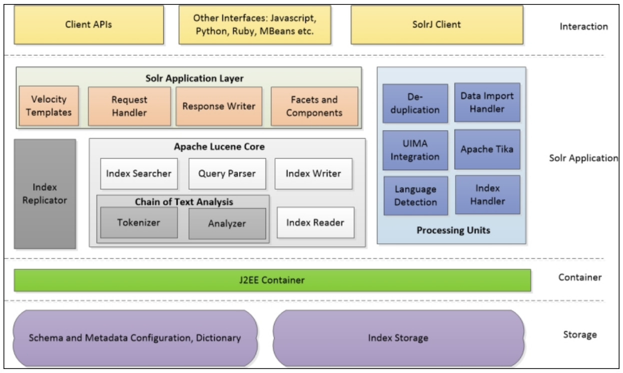


Figure 1: Solr Acrchitecture

SolrJ client is used to interact with Solr search engine. It enables any Java application to talk directly with Solr through its extensive library of APIs. Apache SolrJ is a part of the Apache Solr package.

Apache velocity is a fast open source templates engine, which is used to quickly generates HTML-based frontend.

Index handlers are a type of update handler, handling the task of add, update, and delete function on documents for indexing. Apache Solr supports updates through the index handler through JSON, XML, and text format.

On submitting a query through the User Interface created using Velocity UI, Request Handler assigns the job to appropriate Query Parser which in turn identifies and filters the query.

Index Reader searches the index store and returns the result to response writer.

Response writer formats the output and responds back to the user.

Velocity, which is a Java-based template engine is used to build the ui for the music search. It permits anyone to use a simple yet powerful template language to reference objects defined in Java code. Velocity separates Java code from the web pages, making the web site more maintainable over the long run and providing a viable alternative to Java Server Pages (JSPs) or PHP.

The Velocity Template Language (VTL) is used to incorporate dynamic content into the web pages of the search engine.

Some of the VTL files used are as follows.

- home.vm - Main entry point into templates
- header.vm - top section of page visible to users
- footer.vm - bottom section of page visible to users, includes debug and help links
- main.css - CSS style for overall pages
- query_form.vm - renders query form
- suggest.vm - dynamic spelling suggestions while typing in the search form
- results_list.vm , hit.vm - called for each matching doc, decides which template to use
- facet_fields.vm - display facets based on field values
- VM_global_library.vm - Macros used other templates,
- error.vm - shows errors, if any
- layout.vm - overall HTML page layout
- did_you_mean.vm - hyperlinked spelling suggestions in results

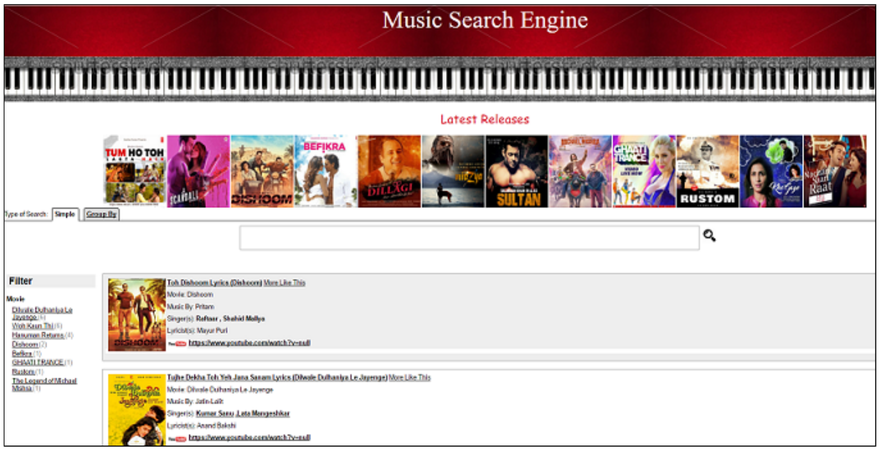
5 Evaluations

5.1 Dataset

The dataset required for populating the search engine is generated by crawling the seed url ‘http://www.hindilyrics.net/lyrics/’

5.2 Evaluation measures

5.3 Results



Dilwale Dulhaniya Le Jayenge (5)
Woh Kaun Thi (6)
Hanuman Returns (4)
Dishoom (2)
Befikra (1)
GHAATI TRANCE (1)
Rustom (1)
The Legend of Michael Mishra (1)

Toh Dishoom Lyrics (Dishoom) More Like This
Movie: Dishoom
Music By: Pritam
Singer(s): Raftaar , Shahid Mallya
Lyricist(s): Mayur Puri
https://www.youtube.com/watch?v=null

Tujhe Dekha Toh Yeh Jana Sanam Lyrics (Dilwale Dulhaniya Le Jayenge) More Like This
Movie: Dilwale Dulhaniya Le Jayenge
Music By: Jatin-Lalit
Singer(s): Kumar Sanu ,Lata Mangeshkar
Lyricist(s): Anand Bakshi
https://www.youtube.com/watch?v=null

Figure 2: Search UI

Figure 2 shows the homepage of the music search engine with its features like latest releases , filters , youtube links, singer profile links , movie thumbnails etc.

5.4 Analysis

The music search engine indexes a set of document i.e the songs with the details and then query Solr to return a set of documents that matches user query.It uses lucene classes to create this index known as Inverted Index. Solr maintains a posting list which holds the mapping of words/terms/phrases and the corresponding places where they occur.The search engine provides high performance indexing.The memory requirement is less as the index size is roughly 20-30% of the size of text indexed.It uses powerful,accurate and efficient search algorithms and pluggable ranking models, including the Vector Space Model and Okapi BM25.