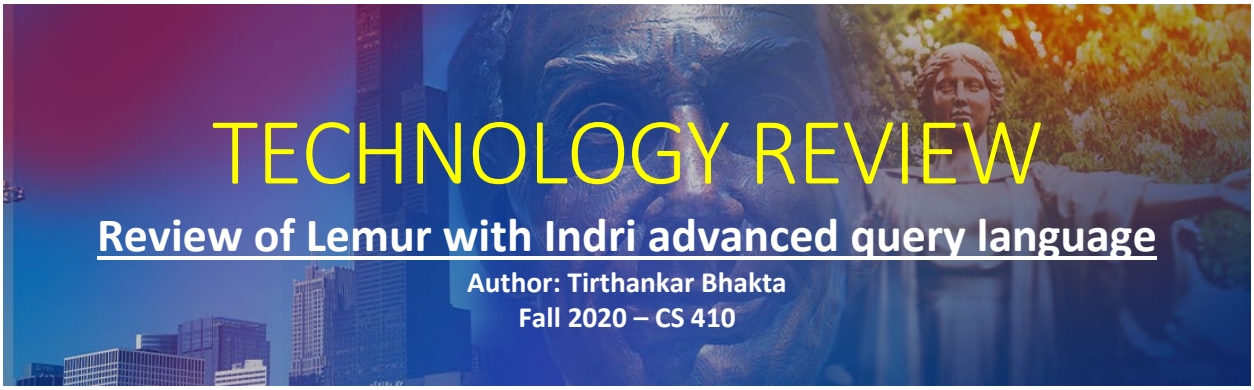


TECHNOLOGY REVIEW

Review of Lemur with Indri advanced query language

Author: Tirthankar Bhakta
Fall 2020 – CS 410

Introduction:

The Lemur Project develops search engines, browser toolbars, text analysis tools, and data resources that support research and development of information retrieval and text mining software; the project is best known for its Indri and Galago search engines, the ClueWeb09 and ClueWeb12 datasets, and the RankLib learning-to-rank library. The software and datasets are used widely in scientific and research applications, as well as in some commercial applications.

In this technology review, we are going to look at the Indri Search Engine which is one of the most notable and component of the Lemur project.

What is Indri:

Indri is a search engine that provides state-of-the-art text search and a rich structured query language for text collections of up to 50 million documents (single machine) or 500 million documents (distributed search) and is supported for both for Linux, Windows and MacOS.

It is an open-source search engine that allows index data or structure documents using simple command line instructions. The Indri search engine can handle large collections of data and can understand various data formats like HTML and XML.

Features of Indri:

- Can make use of multiple document representations
- Explicit term weighting
- Robust query language
- Formally well-grounded
- Highly effective
- Can be efficiently implemented

Indri Retrieval Model:

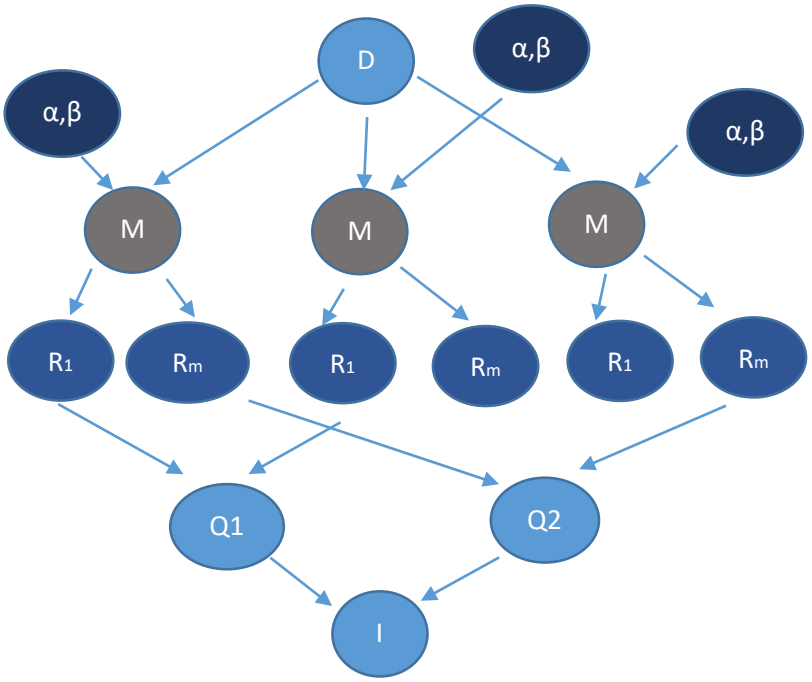
The model is based on a combination of the language modeling (PonteCroft1998) and inference network (TurtleCroft1991) retrieval frameworks. This model was later enhanced and used widely.

Here are some of the features of the model:

- Easily handles phrases (ordered and unordered)
- Can make use of multiple document representations

- Explicit term weighting
- Robust query language
- Formally well-grounded
- Highly effective
- Can be efficiently implemented

The mode can be graphically described as a Bayesian Network diagrammatically represented by:



The network consists of the following types of nodes:

- Document node (D)
- Smoothing parameter nodes (alpha, beta)
- Model nodes (M)
- Representation concept nodes (r)
- Belief nodes (q)
- Information need node (I)

Document Node (D): the document node is a random binary variable represented in a binary vector format where each entry represents presence or absence of a text.

Smoothing Parameters (α, β): these nodes respresent model hyperparameters that correspondes to smoothing framework.

The following types of smoothing are available in Indri:

- Jelinek-Mercer

- Dirichlet
- Two-Stage (ZhaiLafferty2002)

Dirichlet is the default smoothing method used in the ramework.

Jelinek-Mercer:

$$P(r \mid D) = \frac{(1-\lambda)\, tf_{r,D}}{|D|} + \lambda\, P(r \mid C)$$

Dirichlet:

$$P(r \mid D) = \frac{tf_{r,D} + \mu\, P(r \mid C)}{|D| + \mu}$$

Two-Stage:

$$P(r \mid D) = \frac{(1-\lambda)\,(tf_{r,D} + \mu\, P(r \mid C))}{|D| + \mu} + \lambda\, P(r \mid C)$$

Model Nodes (M): Model nodes represent the Language Models. Usually there are multiple model nodes corresponding to different representation of the same model. This allows models to be estimated across different representations and these results can be combined to give a proper output.

Representation Nodes (R): The representation concept nodes are binary random variables that are directly related to the features extracted in the document representation.

Belief Nodes (Q): Belief nodes are binary random variables that are used to combine probabilities within the network. Each belief node corresponds to a different conditional probability table, allowing beliefs to be combined in a number of ways.

Indri Query Language:

The Indri Query Language is a complete structural query language, originally based on the InQuery Query language. The Indri Query Language is used for retrieval and the basic usage can be defined as:

IndriRunQuery/RetEval <parameter_file>

Parameter file includes options like:

- Where to find index
- Main query or sub-queries
- Memory to use
- Formatting options
- Optional parameters

Some basic ways the Indri Query Language can be used are:

- Simple Queries: this can be simple words or a combination of words combined in a logical fashion
- Phrase Matching: searches for a specific phrase. Usually the words are wrapped using the ordered window operator #n (where n is the window size of the number of terms).
- Unordered Windows: #uwN operator to perform a search on terms that occur within a certain window size, but in any order.

Indri API:

Indri API can be used to test or prove efficiency of any retrieval problem.

Basic features of the API are:

- Best for integrating search into large applications
- Has two main objects
- Supports Indri Query Language, XML retrieval, real-time incremental indexing and parallel retrieval.

Summary:

To summarize, the Indri Search component of the Lemur Project is well-recognized and used because of:

Powerful Query Interface

- Supports popular structured query operators from INQUERY
- Suffix-based wildcard term matching
- Field retrieval
- Passage retrieval

Flexible Indexing and Document Support

- Supports UTF-8 encoded text
- Language independent tokenization of UTF-8 encoded documents.
- Parses PDF, HTML, XML, and TREC documents
- Word and PowerPoint parsing (Windows only)
- Text Annotations
- Document Metadata

Package Versatility

- Open source, with a flexible BSD-inspired license
- Includes both command line tools and a Java user interface
- API can be used from Java, PHP, or C++
- Works on Windows, Linux, Solaris and Mac OS X

Scalability and Efficiency

- Best-in-class ad hoc retrieval performance
- Can be used on a cluster of machines for faster indexing and retrieval
- Scales to terabyte-sized collections

Refernces:

1. The Lemur Project official website - http://www.lemurproject.org/indri.php
2. https://sourceforge.net/p/lemur/wiki/The%20Indri%20Query%20Language/