IDEAL Climate Change Project

Somn Kafley, Michael Steele, Samyak Singh  
Software Requirements & Design Report  
CS 4624 Hypertext / Multimedia

TABLE OF CONTENTS

**SOFTWARE REQUIREMENTS**

I**.** [Overview](#h.yr1ugo2xa7v1)…………………………………………………………………………..2

1. [Objectives](#h.wl33g0gx951l)
2. [Business Process / Communication Method](#h.zad2f299hdx6)
3. [User Roles & Responsibilities](#h.6chgw333oxdg)
4. [Interactions with Other Systems](#h.aako31cwz62r)
5. [Production Rollout Considerations](#h.3yv510utyxm9)
6. Timeline of Report
7. [Terminology](#h.mcl5nwhn7gur)

II. [Functional Requirements](#h.ypd5vv5qxcq2)………………………………………….……………..3

1. [Statement of Functionality](#h.gz3gvodx449b)
2. [Scope](#h.qt3w3scylrmc)
3. [Performance](#h.coxbx571lk9h)
4. [Usability](#h.qp3m54t2fmj1)

**DESIGN REQUIREMENTS**

I.[Introduction](#h.2v8f7ih562)……………………………………………………….……...………..5

1. [Overview](#h.y5o66xkyxd1h)
2. [Purpose](#h.y5o66xkyxd1h)
3. [Scope](#h.y5o66xkyxd1h)

II. [System Overview](#h.n8qzdt526mf)………………………………….……………………………..5

III. [Architectural Design](#h.9pbfpvk1blbf)……………………………………………...……………..6

1. [Decomposition Description](#h.l4oggb1qwsrq)
2. [Design Rationale](#h.l4oggb1qwsrq)

IV. [Data Description](#h.sf0718tma5x7)……………………………………………………....………..7

V. [Overview of User Interface](#h.3136jkfc4xzi)……………………………………………………..7

VI. [Screen Objects & Actions](#h.twem4aysh4mh)…………………………………….………………..9

VII. Referenced Images…………………………………………………….……..10

SOFTWARE REQUIREMENTS

# Overview

### Objectives

The primary goal of the IDEAL climate change project is to provide an efficient way for non-technical users to search and access archived tweets and websites related to climate change. The project serves as a tool for researchers who want to utilize the large pool of data behind these websites.

### Business Process / Communication Method

The mode of communication between the engineers and the client is currently via email, due to ease-of-access. Any functional or design requirements are sent to the engineers’ Virginia Tech email. So far, this method of communication has been effective due to the high correspondence-rate between the parties.

### User Roles & Responsibilities

Our primary clients/users are researchers/scientists/engineers who want to utilize the large pool of data access through these websites.The main role of the user is to search tweets and websites by typing in a keyword on the given user-interface. The user is then responsible for using data-extracted from the underlying Solr search engine for various research related to climate-control. Users can also report any search error or usability issues to the design team via email.

### Interactions with Other Systems

Currently, the IDEAL climate change system interacts with an interface built on an Apache search platform called Solr. Solr is an excellent system because of its reliability, scalability, tolerance to faulty queries, distributed indexing, load-balanced querying, automated failover and recovery, and centralized configuration (Apache). Solr controls the search and navigation, but it sits below a user-friendly framework called Blacklight.

### Production Rollout Considerations

On final product-release, the project will be fully-functional and will allow users to search tweets and websites as stated above. Prior to the final release, a semi-final product release will be conducted to test and present to the client. The product is primarily designed to as a research tool for researchers, scientist or environmentalist to see climate change effect studying the climate change tweets and web sites. The goal is to avoid working in a “waterfall” style software development process, and approach quality incrementally.

* 1. Timeline of Reports

|  |  |
| --- | --- |
| Date | Discreption |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Terminology

1. **Apache**: Popular web server software that owns Solr.
2. **Blacklight**: It is an open source ruby on rails engine that provides basic discovery interface. In other words, it is a user-friendly framework that helps interaction between Solr and the user.
3. **Climate Change**: The change in global and regional climate patterns due to the increased levels of atmospheric carbon dioxide produced by fossil fuels.
4. **IDEAL**: Acronym for “Idea, Development, Exploration, Assessment, Long-term study”, IDEAL is a framework for describing the stages of innovation and other interventional procedures.
5. **Java**: A general-purpose programming language well known for class-based and object-oriented approach to problem solving.
6. **Python:** A general-purpose, high-level programming language focused on readability and light syntax.
7. **Ruby**: A dynamic, object-oriented, general-purpose programming language.
8. **Ruby On Rails**: A web application framework written in Ruby which provides data structures for a database, a web service, and web pages.
9. **Solr**: An open source enterprise search platform written in Java from Apache.
10. **Tweet**: A status update from a user limited to 140 characters on Twitter.
11. **Twitter**: A free social networking platform which allows users to follow, tweet, and reach out to other users.
12. **User Interface:** Part of a software with which a human being interacts with.
13. **URL:** Acronym for “Uniform Resource Locator”, which is a reference to a resource on the internet. It consists of two main components (e.g. http), a protocol identifier and the resource (e.g. example.com).

# Functional Requirements

### Statement of Functionality

The software’s top priority is to address every query with pin-point accuracy. As a search based application, it is important that the user is presented with accurate and relevant information. The targeted users are individuals involved in research who may not have a lot of time to dwell on irrelevant search results. The software requires a user-interface which is efficient and fast; a clean and organized way to present the search results, and an accessible platform to be hosted on.

### Scope

The scope of this project consists of several phases with various tasks due at the end of every phase, which will also be tested incrementally.

**Phase 1: Environment setup.**

This is when the Solr server is set up, and Blacklight framework is installed. It also involves downloading dependencies like Ruby, Rails, and Java. After this phase, the environment will be tested so that data extraction and indexing can begin.

**Phase 2: Data extraction from collection.**

The data-extraction phase involves running a Python script to parse, scan, and extract website URLs from relevant tweets related to climate change. The collection of tweets is given to us by the client and resides in a database in the research lab, in Torgersen.

**Phase 3: Data indexing to Solr.**

After the data extraction phase has been completed and properly tested, a second Python script is executed to index all the URLs and tweets to the Solr search platform, from where the user can search by keyword. This is the last “back-end” step before the team starts working on the user-interface.

**Phase 4: User interface design and development.**

The user-interface phase is the last phase whose goal is to create an “easy-to-use” interface for the user. This is an optional phase which will be done only if Phases 1-3 are properly working; this is because the interface provided by Blacklight is usable, but it is possible to have a better one.

### Performance

The software needs to be incredibly fast so that busy researchers are not helplessly wasting precious time. The software doesn’t need to render large images or files which allows it to be swift and secure. Performance should be assessed by how fast the software responds to queries and how fast it can display a large number of results. So, the main factors of performance are time and correct results.

### Usability

The user-interface provided by Blacklight is good. However, the interface can be simplified by removing redundant and unnecessary features. The design goal is to provide a simple experience to achieve good performance. The elements required and interface-features required will be updated and refined as the team approaches Phase 4: User-Interface Design & Development.

SOFTWARE DESIGN

# Introduction

### Overview

The Software Design document is intended to translate the requirements into a software representative. It is important to explain design decisions, architectures used, rational behind architecture choices, and the importance of an efficient user-interface; the design document accomplishes these goals. The document is divided into several sections, each of which informs specific concepts to the interested party.

### Purpose

To describe the architecture and system design of the IDEAL Climate Change project, and describing the design decisions and user-interface.

### Scope

The IDEAL Climate Change project allows users to search keywords related to climate change and get results pointing to websites with relevant information. Detailed information is available in the Section II of the Software Requirements document.

# System Overview

The archive data is stored in XML format and sent via cURL as specified by the SOLR manual. It is then retrieved by Blacklight from the SOLR model and displayed on an interface created with Ruby on Rails; the display settings can be modified through the configuration file which is shipped with Blacklight.

SOLR contains Apache Lucene, which is a search engine written in Java, and serves as the “back-end” for SOLR. Lucene stores a portion of the XML documents and assigns each of these documents a reference id; it creates and stores an inverted index containing keywords from all stored documents. Each keyword in the inverted index contains the document id of each document that stores it. For a visual reference, refer to Section III - Architectural Design.

Depending on the configuration settings, Lucene can store the full text of the document, blurbs from the document (similar to Google), or only the description of the document without the text within. Lucene also stores metadata like the title, date modified, date indexed, URL, etc. By using the inverted index described above, Lucene chooses the document with the largest number of matching keywords. The contents of these files are stored in a custom manner exclusive to Lucene, not in a relational database like MySQL.

# Architectural Design (diagram)

### Decomposition DescriptionHypertextDiagram.PNG

Documents to index must be converted to XML format. Once they have been sent to the index server, Lucene creates index tables that list where keywords are stored in each document. The SOLR server relays the search engine requests to Lucene, which returns the search results.

The Geoblacklight application servers as a GUI and web application. It converts HTTP requests into internal requests that are sent to SOLR. Geoblacklight takes the search results sent in text format from SOLR and displays them in a pretty, formatted way.

Request: HTTP -> blacklight -> Internal request -> SOLR -> Lucene search

Response: Lucene search -> text response -> SOLR -> Geoblacklight -> HTTP

### Design Rationale

The architecture above was selected because of SOLR’s powerful indexing, storing, and accessing efficiency. As opposed to a relational database such as SQL, in SOLR all searchable words are stored in an inverse index, which results in an incredibly faster search. Also, SOLR allows access to internal data structures which a relational database doesn’t. The biggest reason SOLR architecture was chosen was because of the predictability and speed of query speed; a relational database is very dependent on design and use case.

Additionally, our client preferred to work with Blacklight and Solr. Pleasing the client is a top priority, so we prefer to use Solr.

# Data Description

Majority of data this IDEAL Climate Change projects handles will be tweets, URLs, and websites extracted from the URLs. The websites will consist of relevant climate change content based on the user’s search keyword.

The initial source of the data will mainly be organizations (e.g. Oxfam, WWF, Friends of the Earth, Greenpeace, Global Action Plan, etc.), politicians and government (e.g. Al Gore, Bernie Sanders, Ed Miliband, Department of Energy and Climate Change, etc.), news agencies (e.g. The Ecologist, Digg Environment, James Murray, The Climate Desk, etc.), bloggers (e.g. Grist, TreeHugger, Kate Sheppard, Julian Wong, etc.), and campaign groups (e.g. Stop Climate Chaos, 350.org, Plane Stupid, One Climate, Climate Camp, etc.).

# Overview of User Interface (screenshots)

Our interface will display web pages from archived tweets. There are two phases of interface display. First one without displaying archived tweets and another one with detail such as place the tweets or web pages originate, time tweets first originate, a topic of the tweets, and frequency of the tweets. Displaying occur only after adding functionality in Blacklight the search web pages and tweets will display on the interface.

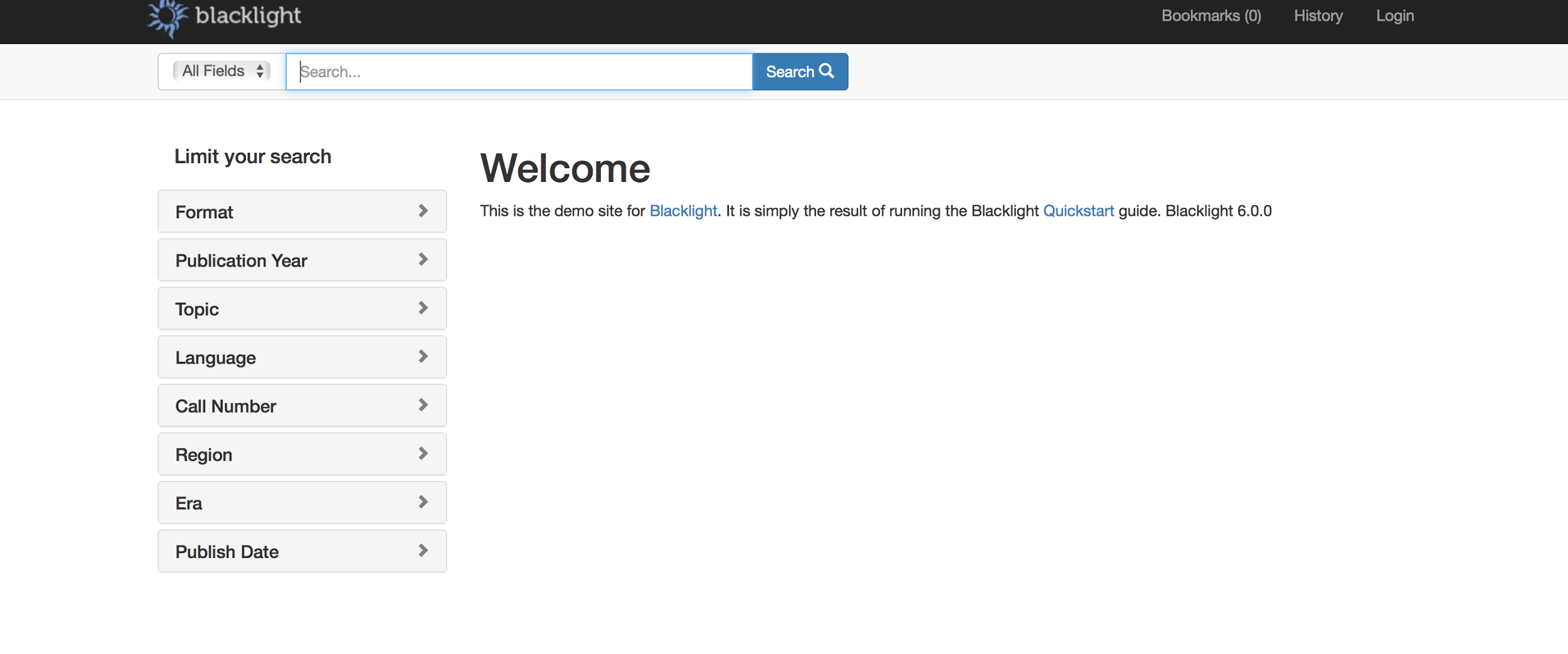
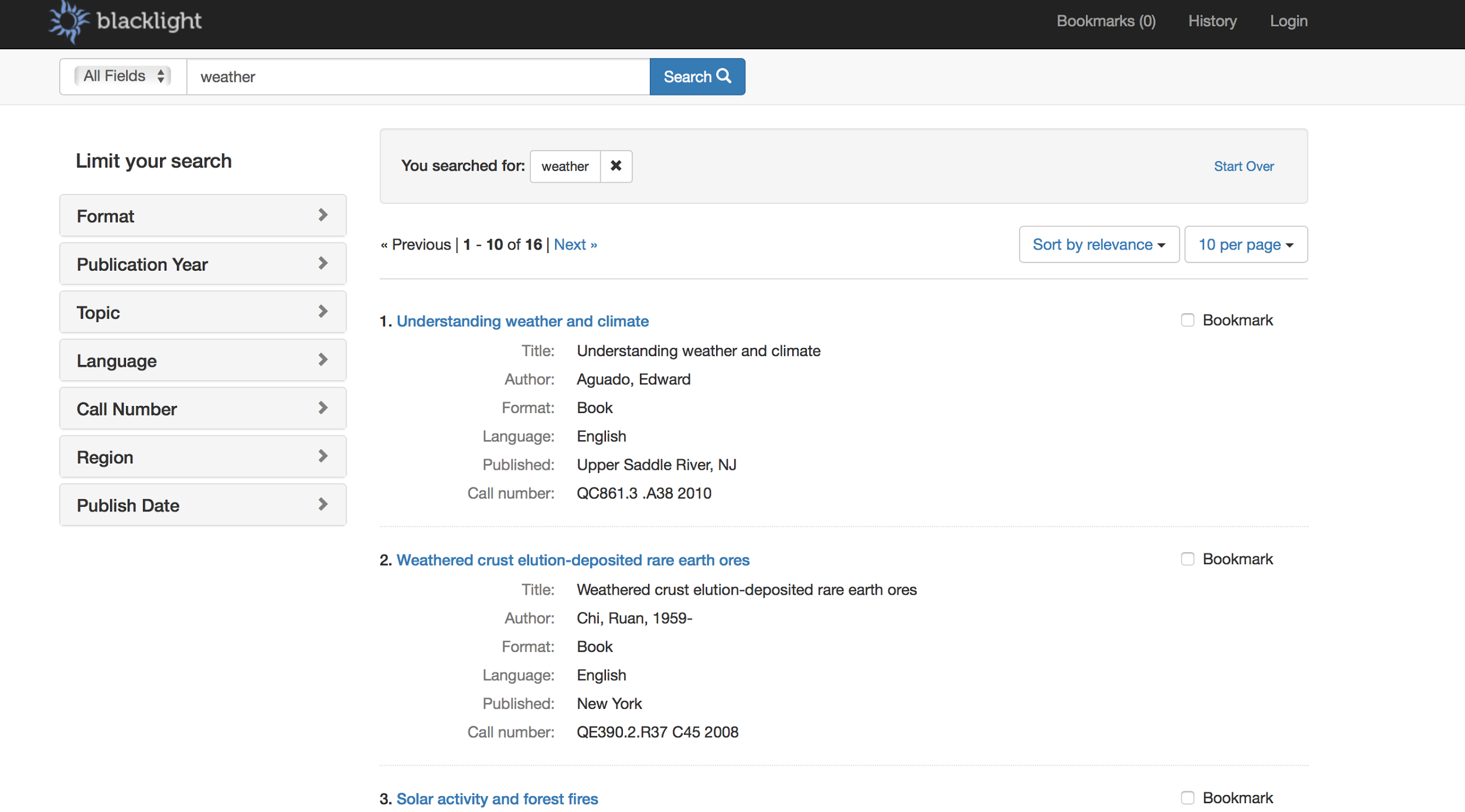


Image: Before search



This is an example of search result display.

# Screen Objects & Actions

# A user can click the search button after typing keywords after which screen objects related to climate change will display in the interface. A user with admin privilege can add content to the website according to their needs. In addition, a user change their search entry anytime and look for different results. By looking into the result a user can correlate trends in climate change webpages & tweets. It will make easier for user to see the change in the frequency of climate change. Thus, through web application users will benefit by being able to search by keyword and by tweet frequency, and will use this information for research.

# 

# Referenced Images

The below images shows the search mechanism that is present in the Lucene search engine within SOLR. The documents are referenced by a document ID in the index table.

