

# **Personalized search of news articles**

Team: Oof1

Masters in Science

University at Buffalo

CSE 535 Information Retrieval (Fall 2014)

Harish Varadarajan

Kaushik Raj Palanichamy

Sankaravadiel Dhandapani

Sathish Kumar Deivasigamani

## Table of Contents

|  |    |
|--|----|
| ABSTRACT: .....                                | 4  |
| SEARCH KEYWORDS:.....                          | 4  |
| INTRODUCTION: .....                            | 4  |
| INDEXING MODULE: .....                         | 4  |
| APPLICATION ARCHITECHTURE: .....               | 5  |
| SOLR CONFIGURATION DETAILS: .....              | 6  |
| IMPORTANT FIELDS: .....                        | 6  |
| Solr Configuration:.....                       | 7  |
| <b>Auto Suggest:</b> .....                     | 7  |
| <b>SpellCheck1:</b> .....                      | 8  |
| <b>SpellCheck2:</b> .....                      | 8  |
| <b>Search Handler:</b> .....                   | 9  |
| LOGIN: .....                                   | 9  |
| <b>Facebook Login:</b> .....                   | 9  |
| <b>Normal Login:</b> .....                     | 10 |
| Anonymous Tracking: .....                      | 11 |
| Personalization: .....                         | 11 |
| Preprocessing of queries: .....                | 11 |
| Post – Processing of queries:.....             | 11 |
| Home Screen .....                              | 12 |
| Login: .....                                   | 13 |
| Search results: .....                          | 14 |
| Complete news article displayed on click:..... | 15 |
| Relevancy Feedback: .....                      | 16 |
| Search Results – User 2: .....                 | 17 |
| Handlers: .....                                | 18 |
| Query Handler: .....                           | 18 |
| SpellCheck:.....                               | 19 |
| Suggest: .....                                 | 20 |
| Dashboard: .....                               | 21 |
| Document Cache: .....                          | 22 |

|                          |    |
|--------------------------|----|
| Work Distribution: ..... | 23 |
| References .....         | 23 |

## ABSTRACT:

The purpose of this project is to build a personalized search engine for three different news articles using Apache Solr. The aim is to apply different concepts learned in the class and implement it.

## SEARCH KEYWORDS:

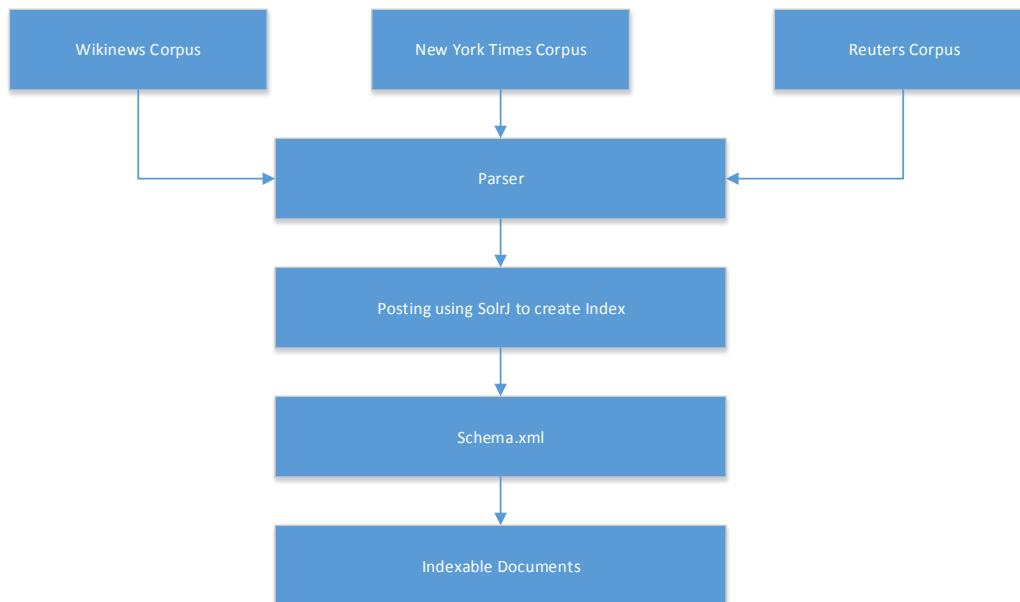
Apache Solr, Apache Lucene, NewYork Times Search, Wiki News Search, Reuters search

## INTRODUCTION:

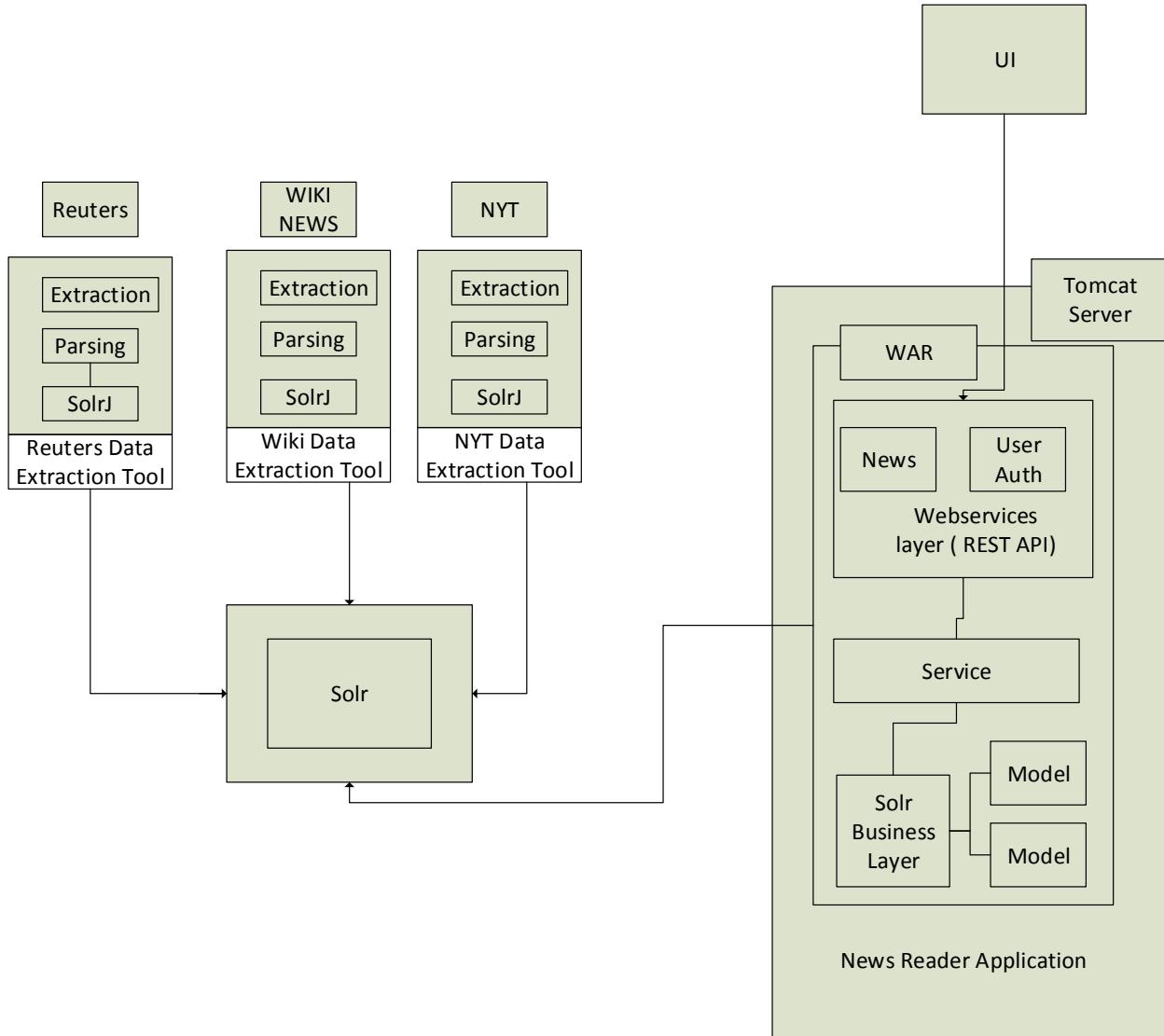
The aim of this project is to give personalized search results for the users based on the user preference and his/her previous searches. We have indexed three different corpus and each corpus has around 10000 articles. We allow user to choose their own preference and also learn from their searches and show results based on their interests and preferences. We also track anonymous user activity and when they login/sign up we map this user activity to the user created. We provide both Facebook OAuth authentication and form authentication for the user.

## INDEXING MODULE:

The main goal of the indexing module is to parse the news corpus and convert it into a common JSON format and post it to the Solr to build index. All the news articles (New York Times, wiki news and Reuters) are converted into a JSON format <sup>[1]</sup>.



## APPLICATION ARCHITECTURE:



## SOLR CONFIGURATION DETAILS:

```
<field name="news_id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
<field name="user_id" type="string" indexed="true" stored="true" required="true" multiValued="true" />
<field name="user_count" type="string" indexed="false" stored="true" multiValued="true" />
<field name="url" type="text_general" indexed="true" stored="true" />
<field name="authors" type="text_general" indexed="true" stored="true" multiValued="true" />
<field name="place" type="string" indexed="true" stored="true" />
<field name="lat" type="float" indexed="true" stored="true" />
<field name="lon" type="float" indexed="true" stored="true" />
<field name="title" type="text_en_splitting" indexed="true" stored="true" />
<field name="published_date" type="string" indexed="true" stored="true" />
<field name="category" type="string" indexed="true" stored="true" multiValued="true" />
<field name="sub_category" type="string" indexed="true" stored="true" multiValued="true" />
<field name="tags_descriptors" type="string" indexed="true" stored="true" multiValued="true" />
<field name="tags_locations" type="string" indexed="true" stored="true" multiValued="true" />
<field name="tags_people" type="text_general" indexed="true" stored="true" multiValued="true" />
<field name="tags_orgs" type="text_general" indexed="true" stored="true" multiValued="true" />
<field name="content" type="text_en_splitting" indexed="true" stored="true" />
<field name="publisher" type="text_en" indexed="true" stored="true" />
<field name="count" type="int" indexed="true" stored="true" />
<field name="subject" type="text_general" indexed="true" stored="true" />
```

## IMPORTANT FIELDS:

**news\_id** - This is the unique identifier of the document.

**user\_id** - This indicates whether the user is clicked or not. It's used in the preprocessing of queries.

**user\_count** – if a user clicks the document, this field is updated. If the user id is already existing, the value will be incremented or the user will be added with value 1. It is used in the post processing of queries

**url** - The URL of the news article.

**authors** – Authors of the news articles. Represented in array

**place** - place of the news article

**lat** – Latitude of the news article

**lon** - longitude of the news article

**title** - title of the news article

**published\_date** - published\_date of the news article. It's represented in YYYY-MM-DD HH:mm:ss format

**category** - category of the news article. This is the highest level classification like politics, technology, sports etc.

**sub\_category** – sub\_category of the news article. This is the next level category in which we mention whether it belongs to football or Asia or Republican etc.,

**tags\_descriptors** - tags\_descriptors of the news article

**tags\_locations** - tags\_locations of the news article

**tags\_people** - tags\_people of the news article

**tags\_orgs** - tags\_org of the news article

**content** - content of the news article

**publisher** - publisher of the news article

**count** - count is the cumulative number of times all the users have clicked on this particular news article.

## Solr Configuration:

### Auto Suggest:

```
<searchComponent name="custom_suggest" class="solr.SpellCheckComponent">
  <lst name="spellchecker">
    <str name="name">suggester</str>
    <str name="classname">org.apache.solr.spelling.suggest.Suggester</str>
    <str name="lookupImpl">org.apache.solr.spelling.suggest.tst.TSTLookupFactory</str>
    <str name="field">autoCorrect</str>
    <str name="buildOnCommit">true</str>
    <float name="threshold">0.0</float>
  </lst>
</searchComponent>
<requestHandler class="org.apache.solr.handler.component.SearchHandler" name="/suggest">
  <lst name="defaults">
    <str name="df">text</str>
    <str name="spellcheck">true</str>
    <str name="spellcheck.dictionary">suggester</str>
    <str name="spellcheck.collate">true</str>
    <str name="spellcheck.onlyMorePopular">true</str>
    <str name="spellcheck.extendedResults">true</str>
    <str name="spellcheck.count">5</str>
  </lst>
  <arr name="last-components">
    <str>custom_suggest</str>
  </arr>
</requestHandler>
```

### SpellCheck1:

```
<searchComponent name="spellcheck" class="solr.SpellCheckComponent">
  <str name="queryAnalyzerFieldType">text_general</str>
  <lst name="spellchecker">
    <str name="name">default</str>
    <str name="field">autoCorrect</str>
    <str name="classname">solr.DirectSolrSpellChecker</str>
    <!-- the spellcheck distance measure used, the default is the internal levenshtein -->
    <str name="distanceMeasure">internal</str>
    <!-- minimum accuracy needed to be considered a valid spellcheck suggestion -->
    <float name="accuracy">0.6</float>
    <!-- the maximum #edits we consider when enumerating terms: can be 1 or 2 -->
    <int name="maxEdits">2</int>
    <!-- the minimum shared prefix when enumerating terms -->
    <int name="minPrefix">1</int>
    <!-- maximum number of inspections per result. -->
    <int name="maxInspections">5</int>
    <!-- minimum length of a query term to be considered for correction -->
    <int name="minQueryLength">4</int>
    <!-- maximum threshold of documents a query term can appear to be considered for correction -->
    <float name="maxQueryFrequency">0.1</float>
    <!-- uncomment this to require suggestions to occur in 1% of the documents-->
    <float name="thresholdTokenFrequency">.0001</float>
  </lst>
  <!-- a spellchecker that can break or combine words. See "/spell" handler below for usage -->
  <lst name="spellchecker">
    <str name="name">wordbreak</str>
    <str name="classname">solr.WordBreakSolrSpellChecker</str>
    <str name="field">autoCorrect</str>
    <str name="combineWords">true</str>
    <str name="breakWords">true</str>
    <int name="maxChanges">10</int>
  </lst>
</searchComponent>
```

### SpellCheck2:

```
<requestHandler name="/spellcheck" class="solr.SearchHandler" >
  <lst name="defaults">
    <str name="df">autoCorrect</str>
    <str name="spellcheck">true</str>
    <str name="spellcheck.collate">true</str>
    <str name="spellcheck.dictionary">default</str>
    <str name="spellcheck.dictionary">wordbreak</str>
    <str name="spellcheck.count">5</str>
    <str name="spellcheck.maxCollations">2</str>
    <str name="spellcheck.maxCollationTries">5</str>
    <str name="spellcheck.alternativeTermCount">5</str>
    <str name="spellcheck.extendedResults">false</str>
  </lst>
  <arr name="last-components">
    <str>spellcheck</str>
  </arr>
</requestHandler>
```

## Search Handler:

```
<requestHandler name="/query" class="solr.SearchHandler">
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <str name="wt">json</str>
    <str name="indent">true</str>

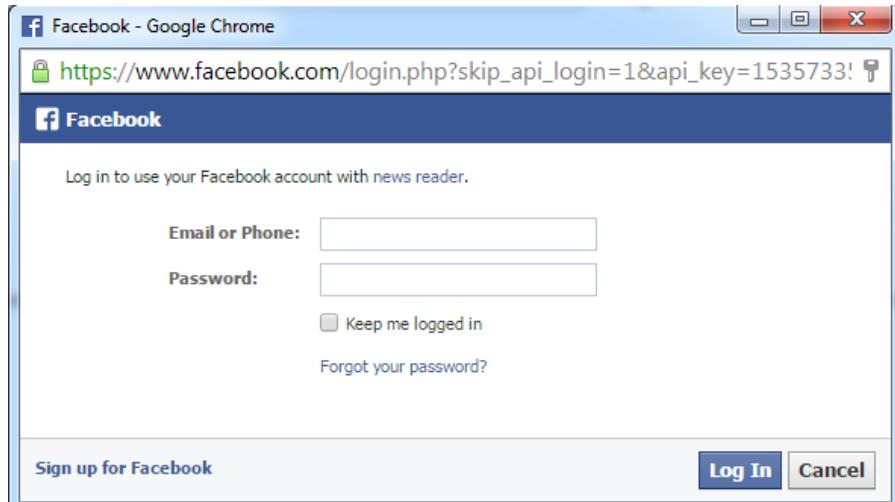
    <str name="defType">edismax</str>
    <str name="qf"> place^7.0 title^10.0 content^3.0 authors^7.0 publisher^7.0 category^0.5</str>
    <str name="df">text</str>
    <str name="mm">100%</str>
    <str name="q.alt">*:*</str>
    <str name="rows">10</str>
    <str name="fl">*,score</str>

    <str name="hl">on</str>
    <str name="hl.fl">content authors title publisher</str>
    <str name="hl.snippets">3</str>
    <str name="hl fragsize">200</str>
  </lst>
  <arr name="last-components">
    <str>clustering</str>
  </arr>
</requestHandler>
```

## LOGIN:

### Facebook Login:

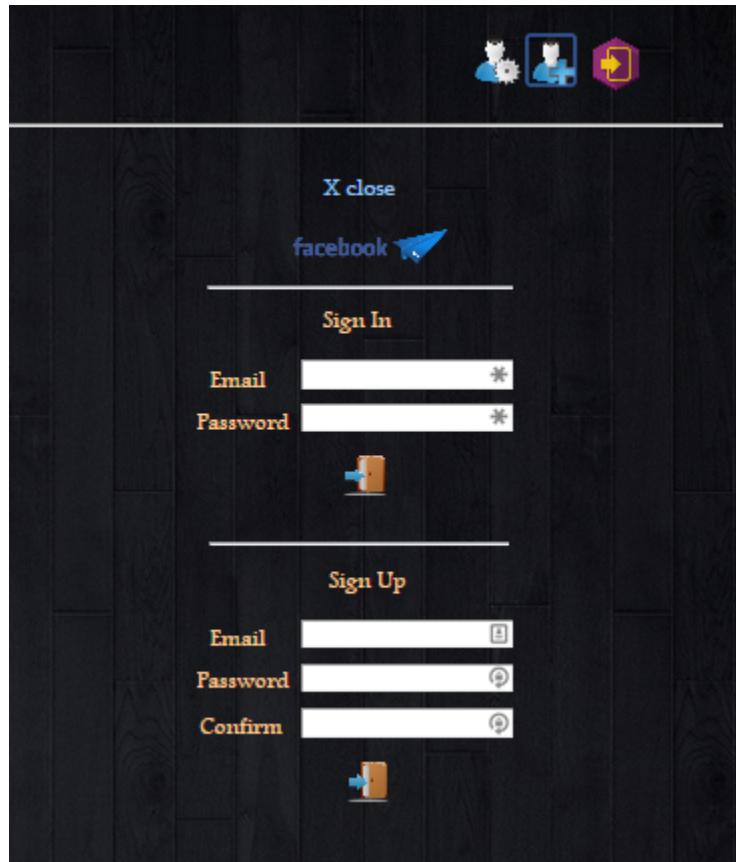
A Facebook app has been created for this project and used for Facebook authentication.



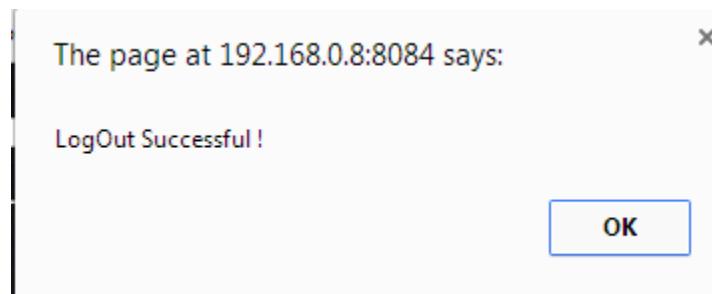
## Normal Login:

Other than Facebook login the user can also login with native form authentication mechanism. Common functionalities like Username creation, authentication is done in java server in- memory. No persistent storage of user details is done. Since we are not using databases, if the server is restarted we would lose all the user information.

“User already exists” and “Invalid login credentials” are the error messages shown for duplicate user names and wrong passwords respectively.



Screen shot of native login screen



Browser Cookie is Cleared during logout

## Anonymous Tracking:

To track the user the user need not be logged into the system. The user can be tracked anonymously. For example, if the user, without logging in starts searching items, we will generate a tracker id (a 36 character guid) and store it in her cookie. All her searches will be tracked against the guid and search history will be stored in the database. Whenever, the user signs in or signs up, we then link the guid with a username and from then onwards, the search history will be stored against the user.

When the user logs out of the system, the cookie is cleared. If a new user starts searching anonymously, a new guid is created to track the user's search history.

## Personalization:

Personalization of search is achieved by preprocessing and post processing queries. Whenever a user clicks a news article link shown in the search page, the id of the news article is captured and stored in the document itself. If user "sam" clicks a news article for the first time, then

1. **"count"** field in the Solr xml will be incremented as it represents the global counter.
2. **user\_id** field will have the user name "sam" added to its array of userids. This is used in preprocessing.
3. **user\_count** field will have the user name "sam" with value 1 added to its array of user counts. This is used in the post processing of results.

At any given point of time, user\_id array length and user\_count array length will be same.

## Preprocessing of queries:

Queries are preprocessed with "AND userid" and if the query returns n results and n less than 10 (10 is the number of results displayed in the page), we fire another query with "AND NOT userid" and append the top 10-n results to the n result set obtained in the first query.

If the both result set does not add up to 10, then it highly likely that the user has misspelled the query. So spellcheck handler is called to get suggestions and in the UI, the suggestion with most search results will be shown as "Did you mean <suggestion>" in hyperlink.

## Post – Processing of queries:

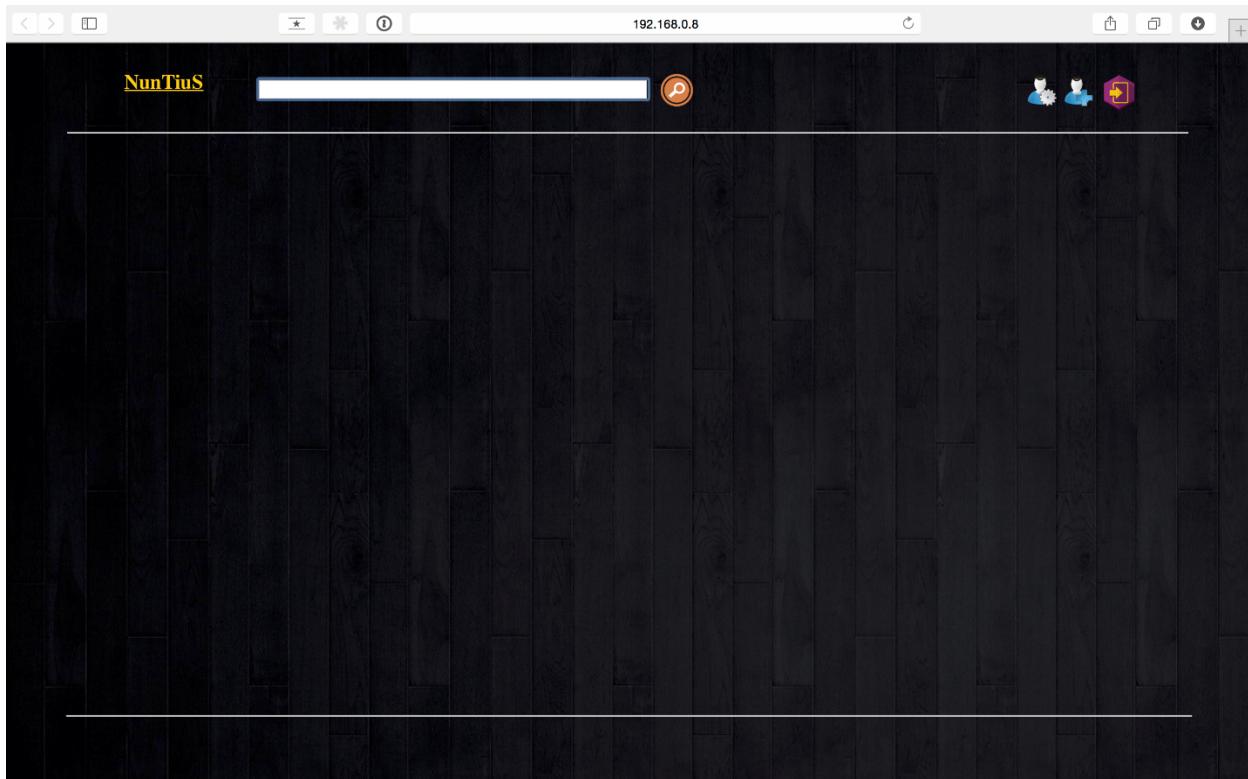
In the post processing of queries we make use of both global counter and the local user counter. Higher weightage is given to the local user hits as it closely represents the user preference thereby user personalization.

$$\text{Finalscore} = (\text{userhits} \times 0.25) + (\text{globalhits} \times 0.1) + \text{Solrscore}$$

The results are sorted in the descending order of the final score.

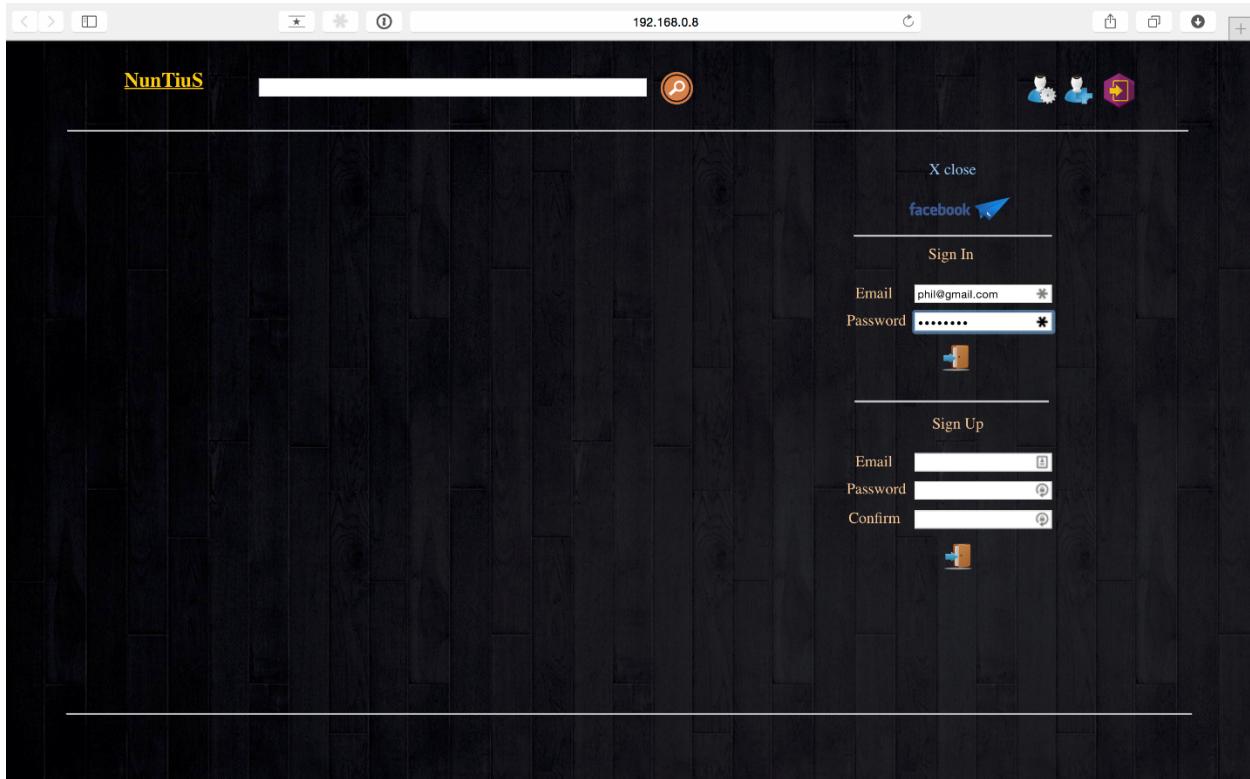
## Home Screen

During page load the IP is sent to the server, a user object is created with the IP and it sends a user primary key as a response, which is stored in the browser cookie. Hence even though the user hasn't logged in, personalization is in process corresponding to his IP.



## Login:

Now when the user sign up is successful, the signed up user id is sent to the server along with the IP. The user object corresponding to this IP is now updated with the user id and password. And all his previous personalization mapped to the user object still remains.



## Search results:

The search query is sent as an AJAX request to the server and the personalized results returned from the server is displayed in the UI in the default order (descending order of the final score).

The first result by default is “**Metro Briefing | New York: Senecas Vote To Rescind Thruway Pact**”

The screenshot shows a web browser window with the address bar containing "Buffalo". The search results are displayed below, with the top result being a link to "Metro Briefing | New York: Senecas Vote To Rescind Thruway Pact". The result includes author information (Staba, David), date (20070420), source (The New York Times), and a snippet of text about the Seneca Nation voting to rescind a 1954 agreement. Below this is another result titled "Despite Foes, Buffalo Museum Makes \$18 Million in Auction", with similar details including author (Kennedy, Randy), date (20070321), source (The New York Times), and a snippet about the museum's sale.

NunTiuS

Buffalo

• [Metro Briefing | New York: Senecas Vote To Rescind Thruway Pact](#)

Staba, David  
20070420  
The New York Times

...Leaders of the Seneca Nation have voted to rescind a 1954 agreement that allowed the New York State Thruway to run through the Cattaraugus Indian Reservation south of *Buffalo*. "The agreement...  
... is declared *null* and void," Maurice A. John Sr., president of the Seneca Nation, said yesterday. The nation's tribal council passed a resolution on Saturday arguing that the agreement, which allowed...

• [Despite Foes, Buffalo Museum Makes \\$18 Million in Auction](#)

Kennedy, Randy  
20070321  
http://www.nytimes.com/2007/03/21/arts/design/21albr.html  
The New York Times

...The first in a series of sales of antiquities from the collection of the Albright-Knox Art Gallery in *Buffalo* made more than \$18 million yesterday at Sotheby's, providing a substantial boost...  
... ever. Since the museum's board decided to sell last fall, the move has drawn fierce criticism from a dedicated group of *Buffalo* residents and deeply divided the museum's supporters. Over...  
... large. Last week, at a meeting forced by the opponents, the museum's members voted 1,224 to 428 in favor of the sales. A last-minute lawsuit filed by the opponents, the

## Complete news article displayed on click:

Once the user clicks on a particular news Title, the entire news content for that particular news is displayed to the user.

In addition to the above, once the user clicks on particular news, the news id is sent to the user and his click on that particular news item is stored for relevancy feedback score.

User Clicked on the news title, "**With DiPietro Back in the Islanders' Crease, Buffalo Folds**"

The screenshot shows a web browser window with a dark theme. At the top, there is a navigation bar with icons for back, forward, search, and refresh. The address bar shows the URL "192.168.0.8". Below the address bar, the page header includes the text "NunTiuS" and "Buffalo" followed by a magnifying glass icon. To the right of the header are three small profile icons. The main content area displays a news article. At the top of the article, there is a link "• < Back to Search Results". The title of the article is "With DiPietro Back in the Islanders' Crease, Buffalo Folds". Below the title, the author is listed as "Higgins, Matt" and the date as "20070415". The location is "BUFFALO, N.Y.". A blue link "http://www.nytimes.com/2007/04/15/sports/hockey/15isles.html" provides the full article URL. The article content begins with a paragraph about the Islanders' franchise goaltender, Rick DiPietro, returning from postconcussion symptoms. It describes the game against the Buffalo Sabres, mentioning several players like Marc-André Bergeron, Ryan Miller, Jason Blake, Tom Poti, and Bruno Gervais, and details the game's progression and outcome.

With their franchise goaltender, Rick DiPietro, healthy again, the Islanders have a new outlook for their Eastern Conference playoff series against the Buffalo Sabres. DiPietro had not played since March 25 because of postconcussion symptoms, but he appeared in top form Saturday night, stopping 32 shots in the Islanders' 3-2 victory. Their first-round series is now tied at one game apiece, with Game 3 on Monday at Nassau Coliseum. DiPietro, who missed 10 games, showed no ill effects from his time off and the Islanders played better in front of him than they had in Game 1 for Wade Dubielewicz, his replacement. Six minutes after the Sabres tied the score at 2-2 in the third period, defenseman Marc-André Bergeron scored the winner on a power play 8 minutes 37 seconds into the period. He took a pass from defenseman Tom Poti and fired a slap shot past Sabres goalie Ryan Miller. "Obviously Ricky was outstanding," Islanders forward Jason Blake said. "Anytime you've got him in the lineup, he definitely gives your club a boost, and certainly a better chance of winning." He added: "He played a great game and it's a testament to his character being out for so long and coming into a playoff game of this magnitude. The way he played was unbelievable." In DiPietro's last start, March 25 against the Rangers, he sustained his second concussion in two weeks. The Islanders played well enough in his absence to clinch the eighth and final playoff seed in the conference. But they appeared overmatched in Game 1 against the top-seeded Sabres, who finished the regular season with the best record in the league. Buffalo dominated in a 4-1 victory Thursday. But from the opening face-off Saturday, it was clear the Islanders were a different team with DiPietro. They scored on their third shot of the game, 3:07 into the game, for a 1-0 lead. A shot by Mike Sillinger glanced off Trent Hunter's skate past Miller. Poti set up the play, finding Sillinger unguarded in the left face-off circle. Sillinger faked a shot before unloading the puck into a crowd in front of Miller. The Islanders continued to apply pressure and grabbed a 2-0 lead at 11:03. With the teams playing four on four, Miroslav Satan beat Sabres forward Daniel Briere cleanly on a face-off. The puck went directly to defenseman Bruno Gervais at the point, and his wrist shot sneaked between Miller's pads. But the Sabres had come back from two goals to win 10 times this season, a franchise record. And for a time they threatened to do so again. Buffalo

## Relevancy Feedback:

Hence, after few clicks on a particular news item, the next time the user searches for the same query that particular news item which the user had already clicked is displayed as the first result.

The first result after personalization is, “**With DiPietro Back in the Islanders’ Crease, Buffalo Folds**”

The screenshot shows a web browser window with a dark theme. The address bar contains the text "Buffalo". The search results page displays two news articles from The New York Times. The top article is titled "With DiPietro Back in the Islanders' Crease, Buffalo Folds" by Higgins, Matt, dated 20070415, from Buffalo, N.Y., with a link to <http://www.nytimes.com/2007/04/15/sports/hockey/15isles.html>. The second article is titled "Buffalo's Vanek Makes Sure Playoff Return Is Not Just Repeat" by Higgins, Matt, dated 20070426, from Buffalo, with a link to <http://www.nytimes.com/2007/04/26/sports/hockey/26sabres.html>.

## Search Results – User 2:

Whereas another user searching for the same query in a different machine has the result ordered in a default order (in case he is searching for the first time) or as per his own personalization (if he had already searched the portal).

Here results are displayed as per his personalization and the first result is, “**Despite Foes, Buffalo Museum Makes \$18 Million in Auction**”

The screenshot shows a web browser window titled "Nuntius" with the URL "192.168.0.8:8084/nr/" in the address bar. The search term "Buffalo" is entered in the search field. The results are displayed in a list format:

- [Despite Foes, Buffalo Museum Makes \\$18 Million in Auction](http://www.nytimes.com/2007/03/21/arts/design/21albr.html)  
Kennedy, Randy  
20070321  
http://www.nytimes.com/2007/03/21/arts/design/21albr.html  
The New York Times  
... large. Last week, at a meeting forced by the opponents, the museum's members voted 1,224 to 428 in favor of the sales. A last-minute lawsuit filed by the opponents, the Buffalo Art Keepers...  
... The first in a series of sales of antiquities from the collection of the Albright-Knox Art Gallery in Buffalo made more than \$18 million yesterday at Sotheby's, providing a substantial boost...  
... was an intricately decorated bronze wine vessel from the Shang dynasty that went for \$8.1 million. A dealer bid for the object on behalf of the Compton Verney museum, a private gallery northwest of London..
- [With DiPietro Back in the Islanders' Crease, Buffalo Folds](http://www.nytimes.com/2007/04/15/sports/hockey/15siles.html)  
Higgins, Matt  
20070415  
BUFFALO, N.Y.  
http://www.nytimes.com/2007/04/15/sports/hockey/15siles.html

## Handlers:

### Query Handler:

The screenshot shows the Apache Solr admin interface at the URL `192.168.0.8:8983/solr/#/collection1/plugins/queryhandler?entry=/query`. The left sidebar shows the navigation menu with "collection1" selected. The main content area displays the configuration for the "/query" handler.

**Handler Configuration:**

- /debug/dump
- /elevate
- /export
- /get
- /query**

**Class:** org.apache.solr.handler.component.SearchHandler  
**Version:** 4.10.2  
**Description:** Search using components:  
query  
facet  
mlt  
highlight  
stats  
expand  
clustering  
debug

**src:** null

**stats:**

| Stat                   | Value              |
|------------------------|--------------------|
| handlerStart:          | 1416885627125      |
| requests:              | 894                |
| errors:                | 0                  |
| timeouts:              | 0                  |
| totalTime:             | 16186.375854       |
| avgRequestsPerSecond:  | 1.2088106849145244 |
| 5minRateReqPerSecond:  | 1.400228590281999  |
| 15minRateReqPerSecond: | 0.7719996935960707 |
| avgTimePerRequest:     | 18.105565832214765 |
| medianRequestTime:     | 16.686456999999997 |
| 75thPcRequestTime:     | 17.698529          |
| 95thPcRequestTime:     | 19.71229525        |
| 99thPcRequestTime:     | 23.063354049999994 |
| 999thPcRequestTime:    | 670.984565         |

## SpellCheck:

The screenshot shows the Apache Solr admin interface at the URL `192.168.0.8:8983/solr/#/collection1/plugins/queryhandler?entry=/spellcheck`. The left sidebar lists various Solr management options like Dashboard, Logging, Core Admin, Java Properties, Thread Dump, and collection1. Under collection1, there are links for Overview, Analysis, Dataimport, Documents, Files, Ping, and Plugins / Stats. The Plugins / Stats section is currently selected. On the right, the SpellCheck configuration is displayed in a tree view under the /spellcheck endpoint. The configuration includes:

- class: org.apache.solr.handler.component.SearchHandler
- version: 4.10.2
- description: Search using components:
  - query
  - facet
  - mlt
  - highlight
  - stats
  - expand
  - spellcheck
  - debug
- src: null
- stats:

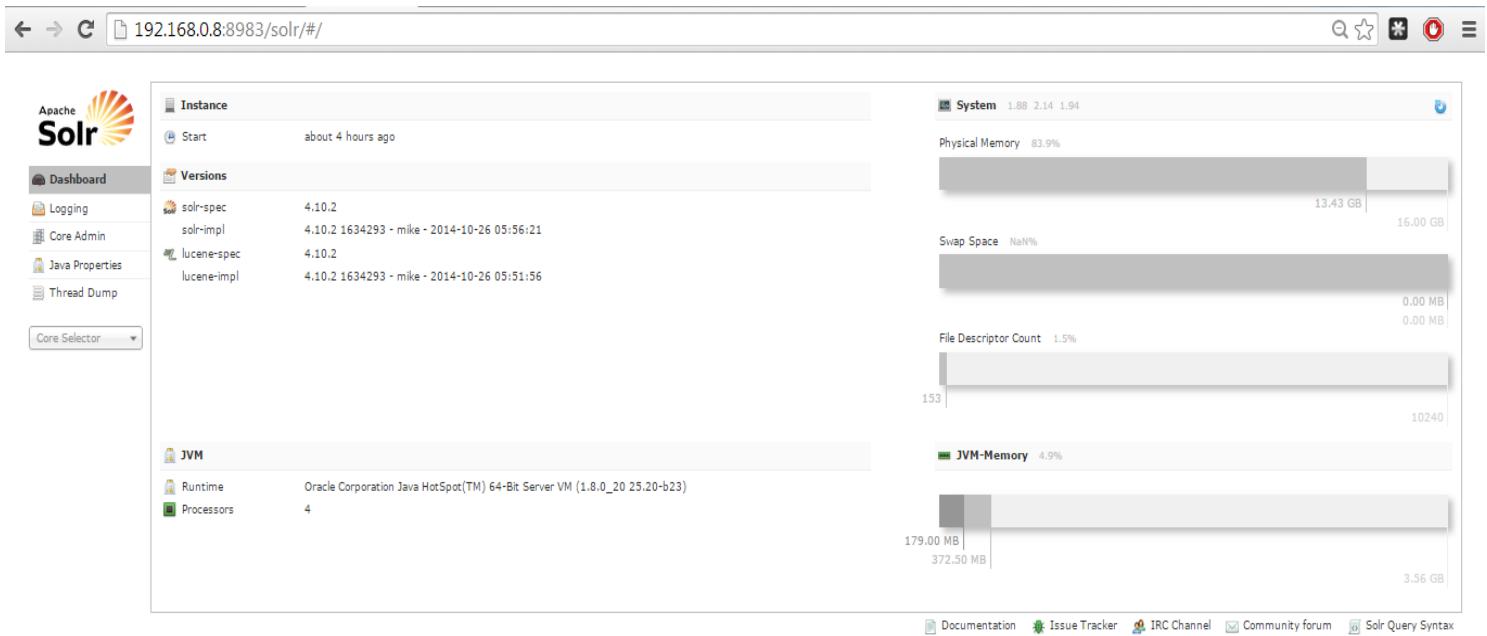
|                         |                     |
|-------------------------|---------------------|
| handlerStart:           | 1416885627127       |
| requests:               | 115                 |
| errors:                 | 0                   |
| timeouts:               | 0                   |
| totalTime:              | 279.258921          |
| avgRequestsPerSecond:   | 0.15549609747462115 |
| 5minRateReqsPerSecond:  | 0.2955283103773949  |
| 15minRateReqsPerSecond: | 0.11716253906734146 |
| avgTimePerRequest:      | 2.4283384434782604  |
| medianRequestTime:      | 2.154225            |
| 75thPcRequestTime:      | 2.821706            |
| 95thPcRequestTime:      | 3.3120503999999986  |
| 99thPcRequestTime:      | 5.39996268          |
| 999thPcRequestTime:     | 5.414979            |

## Suggest:

The screenshot shows the Apache Solr admin interface at the URL `192.168.0.8:8983/solr/#/collection1/plugins/queryhandler?entry=/suggest`. The left sidebar shows the navigation menu with the 'collection1' dropdown expanded. The 'Plugins / Stats' section is currently selected. The main content area displays the configuration for the '/suggest' endpoint. It includes the class (`org.apache.solr.handler.component.SearchHandler`), version (`4.10.2`), and description (`Search using components:`). Below this, a list of components is provided: query, facet, mlt, highlight, stats, expand, custom\_suggest, and debug. The 'src:' field is set to null. A table of statistics follows, showing various request metrics. At the bottom, there is a link to 'Home'.

| stat:                  | value               |
|------------------------|---------------------|
| handlerStart:          | 1416885627127       |
| requests:              | 244                 |
| errors:                | 0                   |
| timeouts:              | 0                   |
| totalTime:             | 57.238553           |
| avgRequestsPerSecond:  | 0.32992260388657874 |
| 5minRateReqPerSecond:  | 0.22082289209453237 |
| 15minRateReqPerSecond: | 0.17440031270289083 |
| avgTimePerRequest:     | 0.2345842336065574  |
| medianRequestTime:     | 0.2130355           |
| 75thPcRequestTime:     | 0.23903             |
| 95thPcRequestTime:     | 0.335465            |
| 99thPcRequestTime:     | 0.8281785500000015  |
| 999thPcRequestTime:    | 1.107195            |

## Dashboard:



## Document Cache:

The screenshot shows the Apache Solr admin interface at the URL `192.168.0.8:9983/solr/#/collection1/plugins/cache?entry=documentCache`. The left sidebar navigation bar includes links for Dashboard, Logging, Core Admin, Java Properties, Thread Dump, collection1 (selected), Overview, Analysis, Dataimport, Documents, Files, Ping, Plugins / Stats (selected), Query, Replication, and Schema Browser. The main content area displays the 'documentCache' configuration and its statistics. The 'documentCache' section shows the following details:

| Parameter   | Value   |
|-------------|---|
| class       | org.apache.solr.search.LRUCache   |
| version     | 1.0   |
| description | LRU Cache(maxSize=512, initialSize=512)   |
| src         | null  |
| stats       | lookups: 16650<br>hits: 5992<br>hitratio: 0.36<br>inserts: 10668<br>evictions: 10156<br>size: 512<br>warmupTime: 0<br>cumulative_lookups: 16650<br>cumulative_hits: 5992<br>cumulative_hitratio: 0.36<br>cumulative_inserts: 10658<br>cumulative_evictions: 10156 |

Below the 'documentCache' section, there are collapsed sections for fieldCache, fieldValueCache, filterCache, perSegFilter, and queryResultCache.

## Work Distribution:

|    |  |                                  |
|----|--|----------------------------------|
| 1  | Architecture design                        | Harish, Kaushik, Sankar, Sathish |
| 2  | Web service layer                          | Sankar                           |
| 3  | Service layer                              | Sankar                           |
| 4  | SolrJ integration layer ( Preprocessing)   | Kaushik, Sathish                 |
| 5  | SolrJ integration layer ( Post processing) | Kaushik, Sathish                 |
| 6  | Utility Development                        | Sathish                          |
| 7  | User Authentication                        | Sathish                          |
| 8  | Models Creation                            | Sathish                          |
| 9  | NYT Data Analysis                          | Sankar                           |
| 10 | NYT Data Extraction Utility                | Sankar                           |
| 11 | Wiki News Data Analysis                    | Kaushik                          |
| 12 | Wiki news data extraction utility          | Kaushik                          |
| 13 | Reuters News Data Analysis                 | Harish                           |
| 14 | Reuters news data extraction utility       | Harish                           |
| 15 | Solr Schema                                | Harish, Sankar                   |
| 16 | Solr Configuration                         | Harish, Sankar                   |
| 17 | Facebook OAuth                             | Harish                           |
| 18 | Application build setup                    | Harish, Kaushik                  |
| 19 | User Interface + AJAX                      | Kaushik, Sathish                 |

## References

<http://terranceasnyder.com/2014/05/user-based-personalization-engine-with-solr/>

<http://java.dzone.com/news/how-write-custom-solr>

<http://www.slideshare.net/LucidImagination/bialecki-andrzej-clickthroughrelevancerankinginsolrlucidworksenterprise-8419715>

<http://www.supermind.org/blog/1059/separating-relevance-signals-from-document-content-in-solr-or-lucene>

<http://www.solrtutorial.com/custom-solr-functionquery.html>