

# **Wargs**

## **News Summarization**

Members:

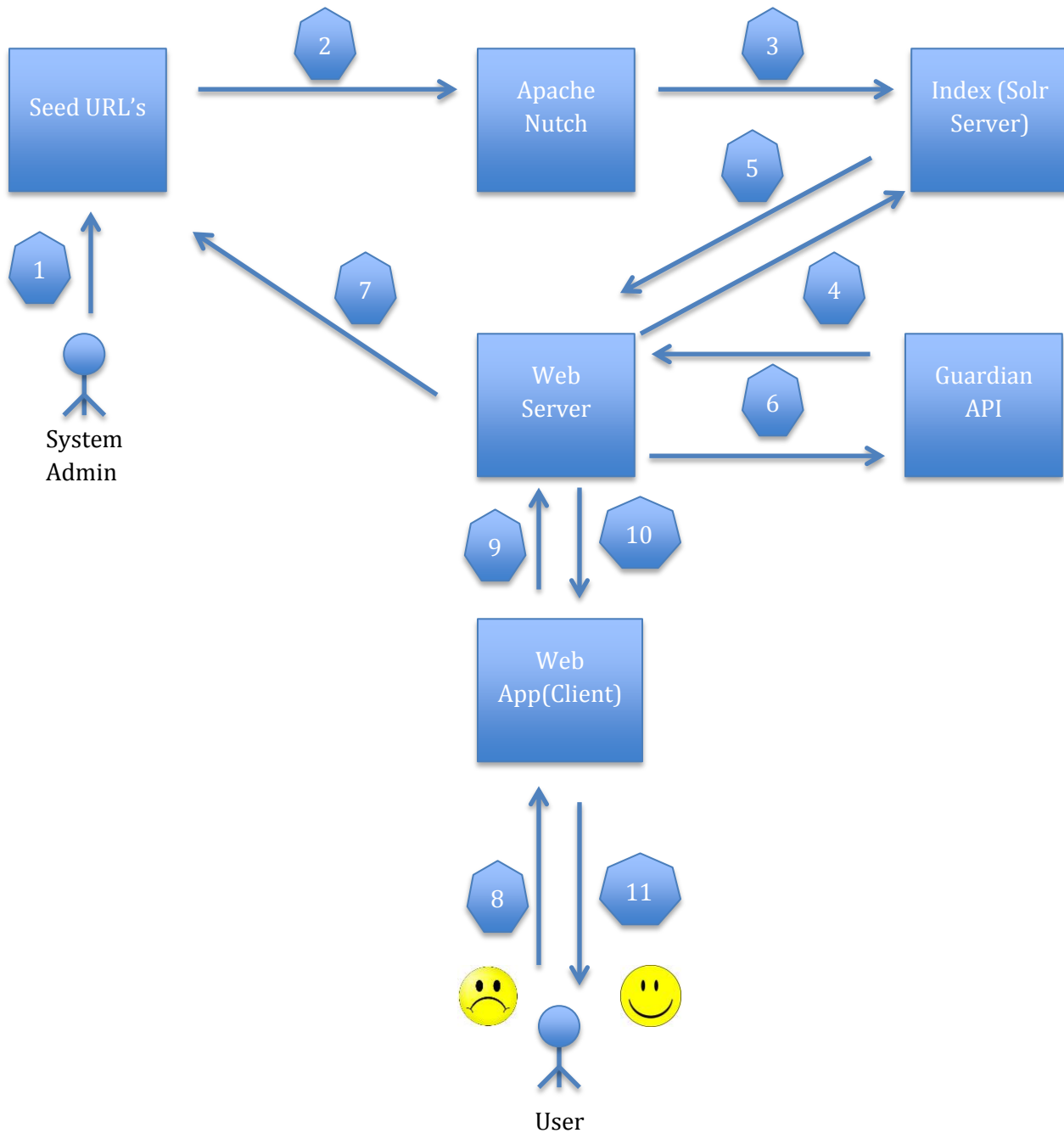
1. Ashish Gupta (agupta28)
2. Girish Suri Venkataraman (gsuriven)
3. Rajaraman Rabindranath (rajaramr)
4. Sai Srinath Sundar (saisrina)

## **Table of Contents**

i.	System Diagram and Description	3
ii.	Features/USP of the System	11
iii.	Configuration Details and Tweaks	15
iv.	Solr Stats	17
v.	Features that we implemented but did not figure into final deliverable	19
vi.	Future Work	20
vii.	Member Contributions	21
viii.	Screenshots	24

## i) System Diagram and Description:

### a. Block Diagram 1 (Using endmemo.com links as the seed pages)



### **Block Diagram 1 Description:**

Step 1: We first prepare a list of seed URL's (start with endmemo.com URL's)

Step 2: These seed URL's are fed to Apache Nutch to crawl through at depth 1.

Step 3: After the crawl is done the pages will be indexed into solr and stored

Step 4: The web server will make a request for the content of these endmemo urls to be sent from the solr index

Step 5: Solr responds with the content of these pages

Step 6: The web server parses the content of the webpages to extract the keywords that are to be sent to "The Guardian" API that in turn returns news articles relevant to these keywords. For ex: if the world event in 'endmemo' was found to be "higgs boson", then upon feeding this to the Guardian API it will return a set of news articles that talk about "higgs boson". We pick the top 5 url's for each keyword as the seed url's for this topic

Step 7: The web server then collates all of these seed url's into one big file and stores it. The older endmemo.com URLs and respective indexed web pages are removed. The crawling and indexing process is repeated for the generated list of URL's from guardian at depth of 3-5 to expand a story/topic. Date extraction is also preformed as part of parsing the web page content from the meta tags.

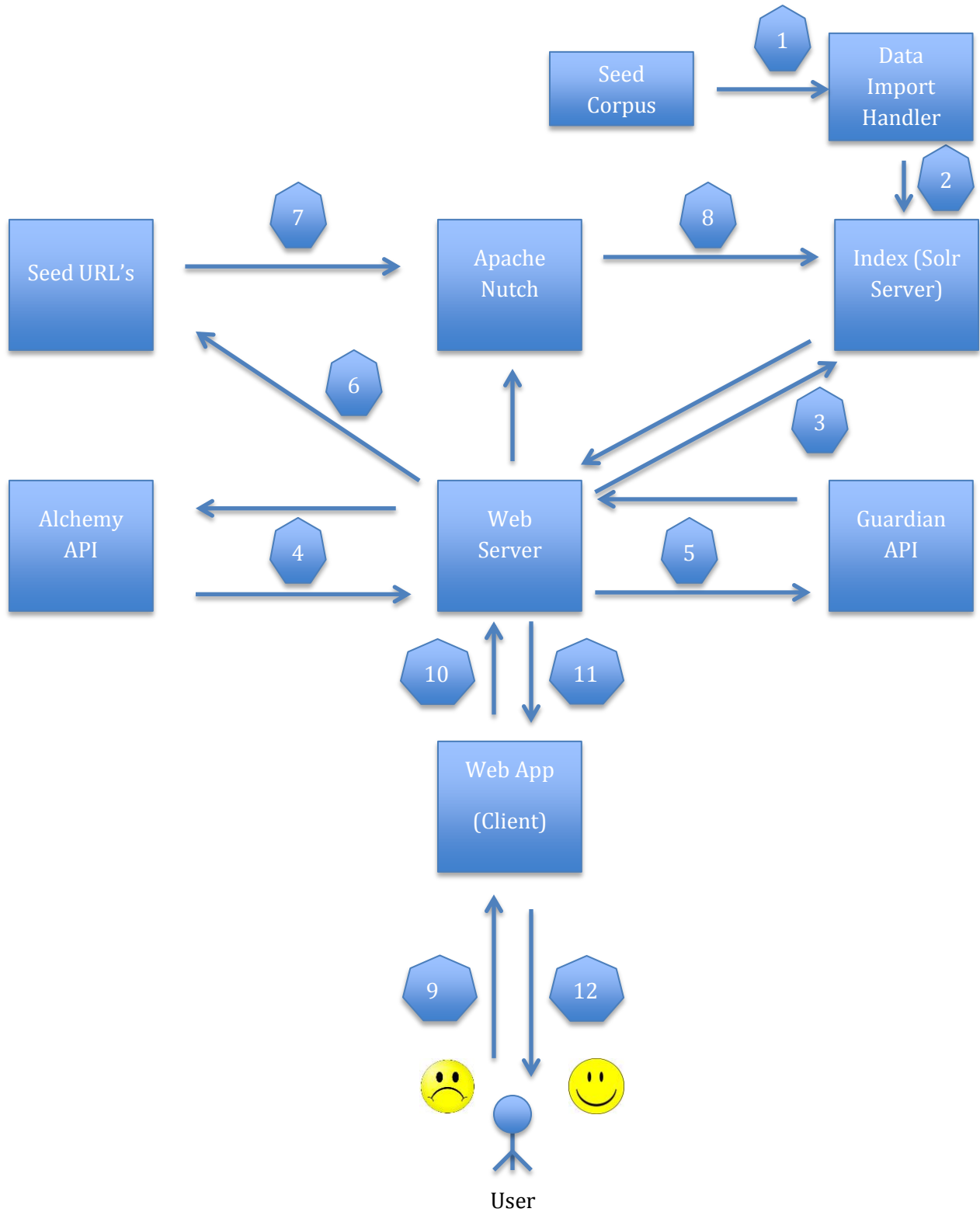
Step 8: The user sends a request to the web app (user interface) in the form of a free text query

Step 9: The web app passes on the query to the web server for further processing

Step 10: The web server talks with the solr index, to check if there are any spelling corrections to be made and returns the most relevant documents with highlighted snippets and clusters that the documents can be grouped into. Also, for each document returned, the solr server looks for 2 most similar documents using the MoreLikeThis feature and returns these as well to be displayed in the view.

Step 11: The web app renders the home view based on the results returned from the web server- which contains the most relevant results with clickable links to the main page, the topical view – which contains a list of documents grouped into common themes or clusters, and the chronological view – which contains the list of most relevant documents ordered in a reverse chronological fashion and for each document it creates links to 2 similar documents

**Block Diagram 2 (Indexing NYT Corpus through DataImportHandler)**



## **Block Diagram 2 Description: (System Architecture if we use NYTimes or RCV Corpus)**

Step 1: Seed corpora like the NYTimes Corpus are sent through data import handler that converts the files into solr indexable format

Step 2: DataImportHandler interfaces with solr to index all the files

Step 3: The web server communicates with solr to get the news articles from the seed corpus

Step 4: These articles are sent to AlchemyAPI for keyword extraction which are stored by the web server

Step 5: The keywords are then queried on the Guardian API to get the 5 most relevant news articles

Step 6: These URLs are aggregated into one big file and stored for crawling in the future

Step 7: Seed URL's are fed to Apache Nutch which does the crawl and stores the parsed content and extracts the published date of the article.

Step 8: This parsed content is indexed into Solr through the Nutch-Solr interface

Step 9: The user sends a request to the web app (user interface) in the form of a free text query

Step 10: The web app passes on the query to the web server for further processing

Step 11: The web server calls the query on solr index, to check if there are any spelling corrections to be made and returns the most relevant documents with highlighted snippets and clusters that the documents can be grouped into. Also, for each document returned, the solr server looks for 2 most similar documents using the MoreLikeThis feature and returns these as well to be displayed in the view.

Step 12: The web app renders the home view based on the results returned from the web server- which contains the most relevant results with clickable links to the main page, the topical view – which contains a list of documents grouped into common themes or clusters, and the chronological view – which contains the list of most relevant documents ordered in a reverse chronological fashion and for each document it creates links to 2 similar documents

## **Detailed Description**

### **Source of the News Articles**

We decided to not go with the given News Corpus because of the lack of freshness in the content and difficulty in crawling NYT.

We crawled ~1lakh pages from TheGuardian and TheHindu but only ~21K could make it to the final corpus because of continuous changes in the schema and some mistakes we made while we were figuring out how Nutch-Solr work together.

### **Smart Crawling using Apache Nutch**

We used Apache Nutch for crawling and indexing the related news articles from the web. Using Nutch out of the box with some seed URL's, blindly crawls the entire web unless otherwise specified. We researched and found out a way to restrict the crawl domain and control the queue by filtering out bad URL's.

### **Preparing the Seed URL's and controlling the queue**

The seed URL's were generated using a website [www.endmemo.com](http://www.endmemo.com) which provides hot events/trends for a particular year. We first crawled the endmemo.com 2012-2014 web pages at depth 1 with 'bad URL' filtering and filtering out the information we did not need. These pages were then indexed in SOLR. Using a ruby script, these pages were retrieved as result of the query, appropriate filtering was then performed to filter out unwanted information to get keywords which were then used with TheGuardian News API's to get URLs of the relevant news which were used as seed URLs for crawling to expand a given topic/story/keyword.

Another approach which was tried but not used due to the lack of freshness in the content was using the seed corpus from NYT which has been explained in block diagram2. We also faced many issues while crawling the NY times web pages as a get call to the web pages using nutch was returning 301 ( permanently moved status code) and we were not able to overcome this issue.

The domain of crawling was restricted to "The Guardian" and "The Hindu". This ensured focused crawling with a max depth of 5 so that the crawler sticks to relevant news and doesn't go offtrack . All the webpages on Guardian/The Hindu had some links which were not news stories for e.g. (<http://www.theguardian.com/world>,<http://www.theguardian.com/fashion> etc. ). These pages, though not news stories contained some keyword and topics, which when indexed were hampering the search results in SOLR. We had to filter out these

URL's so that they do not appear in the crawler queue. We found that the URLs of news articles follow a pattern which was exploited by us to crawl only news stories.

Filters used:

+^http://www.theguardian.com/.+/[0-9]{1,4}/

e.g. <http://www.theguardian.com/us-news/2014/dec/01/obama-white-house-summit-ferguson>

+^http://www.thehindu.com/.+/.+/.+

e.g. <http://www.thehindu.com/news/national/former-maharashtra-chief-minister-ar-antulay-passes-away/article6654510.ece?homepage=true>

Major world events over the past 3 years have been crawled by hitting the The Guardian API and generating a list of seed pages.

### **Date Extraction**

One of the main requirement of the project was to show the chronological summarization view of a given query. Due to this, date extraction from news article became a key task.

Most of the news articles on the web these days have a article published date/time in the meta tag of the web page. We exploited this and extracted the news article date from the meta tags of web pages of TheGuardian and TheHindu.

An open source extractor plugin released by BayanGroup has been used for parsing out the published date of all the news articles based on their meta tag properties. We add this plugin to the list of plugins that nutch uses while crawling and parsing data

TheGuardian

```
<meta property="article:published_time" content="2014-12-01T23:04:11.000Z"/>
```

TheHindu

```
<meta property="article:published_time" content="2014-12-02T11:12:07+05:30" />
```



## Indexing into SOLR

The web pages were indexed into SOLR using the Nutch-SOLR interface in which the URL of the web page used as the key for duplicate web page identification and deletion.

## Topical Summarization View

We used Carrot's Lingo Clustering Algorithm to identify topics for a given result set. Lingo Clustering algorithm first tries to identify the labels from the term-document matrix using LSI and then assigns the document to the labels based on whether the document contains the label words.

Why we did not go with K-means or Suffix-Tree Clustering?

- K-means : This produces very short labels and produces non-overlapping clusters . We found out that some of the short labels were meaningless.
- STC clustering (Suffix Tree Clustering) : Though this produces overlapping clusters, the number of clusters produced were less , the smaller clusters were rarely highlighted and the cluster labels were not descriptive enough.

Lingo Clustering Algorithm produces reasonable number of clusters with descriptive clusters. We found out that the clustering does not work well if the number of articles for a query are less. Approximately 50 noise-reduced articles produced better clusters.

The image below shows the topics for a given query “box office results for interstellar”. Some of the topics produced are highly relevant while some of them make no-sense due to lot of noise present in the articles. Noise corresponds to unwanted text present on the web pages which is not related to the query being answered.

Topics Related to box office results for interstellar

Events Related to box office results for interstellar

Showing Results for : [box office results for interstellar](#)

Total Search Results : 17952

Interstellar Voyage through the Oculus Rift Wormhole		Widest-ever IMAX Rollout Anticipated Film the Guardian		Delivers Film the Guardian		Matthew McConaughey and the Cast of Interstellar	
Total	Hopes	Catherine Shoard and Henry Barnes	Interstellar and Sci-fi's Obsession with Americana Popular	Space Odyssey	Other Topics		

We found out that, one way to reduce noise apart from initial filtering (special char filter, stemmer etc) is to, instead of providing the whole content of a news articles , use the summary of the news articles for clustering. This not only improved the quality of labels but also speeds up the clustering process. Summary has the keywords of the news while removing unwanted/unrelated text. We used a total of 5 summaries of each document with a length of 500 characters per summary.

```
<str name="carrot.produceSummary">true</str>
```

```
<str name="carrot.summarySnippets">5</str>
```

We used the title and a summary of content field of each document for clustering.

```
<str name="carrot.title">title</str>
```

```
<str name="carrot.snippet">content</str>
```

The articles in each cluster were sorted by the score in the cluster and shown in a high-score document first ordering in the web-ap

### **Chronological Summarization View**

We realized the chronological view over and above the template for our webpage. Given that we were extracting dates from the crawled pages. The task entailed sorting the retrieved results in a chronological order, finding the difference in dates between current date and the articles date and using that as metric to sort the articles. Some of the problems we faced in making this work were related to the date field being of different formats. Ruby's date libraries were used to parse these varied date formats and then a simple comparator was applied on the results to have them sorted.

We used the top 10 results of the user query to summarize the "user query"/story over a given period of time. Post sorting the documents freshness, we proceeded to display the date variables next to the articles along with the time lapsed between article and the current date. This helped us get a view that shall show to the user of the application how the story developed over a period of time

## **ii) Features/USP of the System**

1. Clustering – Topical Analysis ( Explained above )
2. Related News Suggestions - MoreLikeThisHandler to find similar documents
  - a. We get the interesting terms for a single document using the mlt.InterestingTerms parameter of the handler and fire another query using these terms to find similar documents
  - b. This is done for each document returned
  - c. After some tuning we found the following parameters to produce the optimal results:
    - i. Mlt = true (this enables more like this feature of solr)
    - ii. Mlt.fl = content ( sets the content field of each document to be the base on which the interesting terms have to be collected)
    - iii. Mlt.interestingTerms = detail
    - iv. Mlt.count = 2 (the number of similar documents to be returned, in this case 2)
    - v. Mlt.match.offset = 10 ( this says that for the first 10 documents returned perform a MoreLikeThis and find similar documents based on the interesting terms)
3. Spelling Correction
  - a. SpellCheckComponent of solr has been integrated to generate suggestions based on the closest terms present in the index based on the Levenshtein distance
  - b. SpellCheck has many default parameters, below are some of the more interesting ones that we tuned to obtain better results:
    - i. Breakwords = true (splits a multi-term query and finds suggestions for each term in query)
    - ii. Combinewords = true (also look at terms combined with each other and try to match it with shingles in our index)
    - iii. maxResultsForSuggest = 5 (if < than 5 documents are fetched by the query then look for spelling corrections even if the query is phrased correctly)
    - iv. collate = true (look for spelling corrections for each token individually)
    - v. count = 1 (return only 1 suggestion; that is the best one according to the edit distance)

We used the content field as a dictionary for spelling correction. Single word spelling correction is simply done using edit distance.

e.g. Eboala -> ebola

For multi word spell correction, a phrase was broken down into words and then a spelling correction was looked for each word using edit distance. Then using the best results of the edit distance measure of each term, a collation query was fired . The collation which produced more hits was treated as the best spelling correction for a multi term query.

e.g. eboala treatmen -> eboala treatment

```
"collation",
  [
    "collationQuery",
    "ebola treatment",
    "hits",
    1659,
    "misspellingsAndCorrections",
    [
      "eboala",
      "ebola",
      "treatmen",
      "treatment"
    ]
  ],
"collation",
  [
    "collationQuery",
    "ebola threaten",
    "hits",
    1252,
    "misspellingsAndCorrections",
    [
      "eboala",
      "ebola",
      "treatmen",
      "threaten"
    ]
  ]
]
```

#### 4. Highlighting

- a. HighlightingParameters feature of solr has been used to generate snippets on the content of the article
- b. We tweaked the following parameters of the highlighting feature of solr:
  - i. Fl = content ( perform highlighting on the content field of each article)
  - ii. Simple.pre = <b> and simple.post </b> (highlight with bold tags the terms that users search for)
  - iii. hl.fragSize = 500 (a fragSize of 500 was found to be the best fit as the content was long enough to understand the context of the article and term searched)
  - iv. hl.snippets = 3 ( maximum of 3 snippets will be produced while highlighting)

#### 5. Multi Source News articles (The Guardian, The Hindu)

- a. Articles from The Guardian and Hindu have been aggregated for similar topics. These topics include world events, sports and politics etc.

#### 6. Shingling for Accurate two term query matching

- a. We have created shingles so that two term queries are matched first with the shingle as well as individually to give more weightage to phrases in the document

#### Why shingles?

Single term queries produce good result without any kind of weighting scheme. Multi term queries fail to give relevant result because of many reasons. It might happen that the each term in the muti-term occurs together in the document but not within a proximity of 1. The documents in which it occurs together with a proximity of 1 are more relevant than in which it occurs at different positions and should get a higher score.

To handle this we created 2-shingles of the content field and indexed them.

At query time, the user entered text was shingle filtered which produced single terms as well as 2-shingles which acted as part of the query vector which were then used for scoring.

## 7. Stanford NER

News articles are full of named entities. They refer to places, people, organizations and a host of objects which could range across many domains Sports(Cricket, Football etc.), Politics(Parliament, European Union etc.). The general idea was to extract these entities, in abundance in a news article, and have them tag the article. Stanford NER is a free to use java library which enables extraction of named entities from a piece of text. The link from “searchbox” listed below was used to integrate the library with Solr. A “ner” request handler was created in solrconfig.xml and Solr was configured to have the handler added to the chain of processes that are executed when a document is indexed.

We found that adding “ner” at the indexing time was expensive in terms of time and produced sketchy results (example “United States of America” was extracted out as “United”, “America”). The feature was abandoned due to lack of correctness. However on further investigation we found that when NER is applied to the retrieved results to extract entities it did a great job. We realized that at index time NER was being applied post tokenization and therefore was of no help as with lack of text NER was unable to recognize named entities in their entirety. We however refrained from adding this feature due to lack of time and delays caused by some of the problems that were table place at indexing time.

<http://www.searchbox.com/named-entity-recognition-ner-in-solr/>

## 8. Edismax parser – for multi term queries and normal parser for single word queries

One term queries were handled using standard lucene parser. For 2 term queries , we used the edismax parser with boosting. We boosted the title field for multi term queries to produce better results.

Title Boosting: title^2 content^1.0

### **iii) Configuration Details and Tweaks**

#### **1. Schema Fields Used:**

- a. Id (type string): Stores the URL of the article. It is the unique identifier for each document
- b. Title (type text\_general): Stores the headline of the article
- c. Content (type text\_general): Stores the content of the article
- d. Text (type text\_general): Copyfield that stores the content, title and url of the article as multivalued terms
- e. publishedDate (type text\_general): Stores the published date of the article in its raw UTC format (yyyy-mm-ddThh:mm:ssZ)
- f. updatedDate (type typeDate): Copyfield that takes the content of the publishedDate and is used for parsing yyyyymmdd date for our chronological view
- g. url (type url): Also stores the URL of the article. Currently, not used anywhere
- h. Add the fields from NER – person, organization, location,

Since the NER plugin required us to provide the schema with fields that shall get extracted out and store in Solr. We have to provide the names of the entities namely Organization, Person, Location and Object. These fields were stored and not indexed as one can imagine the kind of problems that would be created if we were to index the entities extracted from an article again along with the source content.

#### **2. Field Types used and their analyzers/filters:**

- a. text\_general
  - i. CharFilters:
    1. solr.PatternReplaceCharFilterFactory – This is to remove junk content that is crawled and indexed by nutch.
    2. Solr.StandardTokenizerFactory – To tokenize the content
    3. Solr.ShingleFilterFactory – To create shingles of two words in size and add to the index. This was used for phrase query boosting
    4. Solr.StopFilterFactory – to remove stopwords from the content
    5. Solr.EnglishPossessiveFilterFactory – to remove 's from possessives
    6. Solr.ASCIIFoldingFilterFactory – to convert non-ASCII characters to ASCII

7. Solr.EnglishMinimalStemFilterFactory – to stem the tokens
    8. Solr.ClassicFilterFactory – removes apostrophes and dots from tokens
    9. Solr.LowerCaseFilterFactory – converts token to lowercase
  - b. typeDate (custom text field)
    - i. Solr.WhitespaceTokenizerFactory – tokenize by whitespace
    - ii. Solr.PatternReplaceFilterFactory – used to trim the UTC format into yyyyymmdd format as need for sorting on the published date
  - c. url –
    - i. Solr.StandardTokenizerFactory – to tokenize the content
    - ii. Solr.LowerCaseFilterFactory – to convert to lowercase
    - iii. Solr.WordDelimiterFilterFactory – split words that are actually two separate words
  - d. String –
    - i. No Filters
3. Similarity Score :
  - a. After experimenting with the BM25Similarity Factory, DFRSimilarity Factory and LMDirichlet similarity factory and not seeing major changes in rankings of the documents , we decided to go with DefaultSimilarity Factory which is based on VectorSpaceModel.



## iv) Solr Stats

### Index Size



- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump

collection1

#### Overview

- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats

#### Statistics

Last Modified: about 2 hours ago  
Num Docs: 21017  
Max Doc: 21017  
Heap Memory: 2330632  
Usage:  
Deleted Docs: 0  
Version: 1275  
Segment Count: 1  
Optimized: ✓  
Current: ✓

#### Instance

CWD: C:\solr-4.10.2\exampleNutch  
Instance: C:\solr-4.10.2\exampleNutch\solr\collection1  
Data: C:\solr-4.10.2\exampleNutch\solr\collection1\data  
Index: C:\solr-4.10.2\exampleNutch\solr\collection1\data\index  
Impl: org.apache.solr.core.NRTCachingDirectoryFactory

#### Replication (Master)

	Version	Gen	Size
Master (Searching)	1417498181877	249	777.02 MB
Master (Replicable)	1417498181877	249	-

#### Healthcheck

Ping request handler is not configured with a healthcheck file.

#### Admin Extra

### Cache Details



- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump

collection1

#### Overview

- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats
- Query
- Replication
- Schema Browser

#### CACHE

- CORE
- HIGHLIGHTING
- OTHER
- QUERYHANDLER
- QUERYPARSER
- UPDATEHANDLER
- Watch Changes
- Refresh Values

#### documentCache

class: org.apache.solr.search.LRUCache  
version: 1.0  
description: LRU Cache(maxSize=512, initialSize=512)  
src: null

stats:

lookups:	691
hits:	633
hitratio:	0.92
inserts:	68
evictions:	0
size:	68
warmupTime:	0
cumulative_lookups:	691
cumulative_hits:	633
cumulative_hitratio:	0.92
cumulative_inserts:	58
cumulative_evictions:	0

#### fieldCache

#### fieldValueCache

#### filterCache

#### perSegFilter

#### queryResultCache



- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump
- collection1
- Overview
- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats
- Query
- Replication
- Schema Browser

## CACHE

- CORE
- HIGHLIGHTING
- OTHER
- QUERYHANDLER
- QUERYPARSER
- UPDATEHANDLER
- Watch Changes
- Refresh Values

### documentCache

### fieldCache

### fieldValueCache

### filterCache

### perSegFilter

### queryResultCache

class: org.apache.solr.search.LRUCache  
version: 1.0  
description: LRU Cache(maxSize=512, initialSize=512)  
src: null

stats: lookups: 18  
hits: 12  
hitratio: 0.67  
inserts: 7  
evictions: 0  
size: 7  
warmupTime: 0  
cumulative\_lookups: 18  
cumulative\_hits: 12  
cumulative\_hitratio: 0.67  
cumulative\_inserts: 6  
cumulative\_evictions: 0

[Documentation](#) [Issue Tracker](#) [IRC Channel](#) [Community forum](#) [Solr Query Syntax](#)

## Average Response time



- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump
- collection1
- Overview
- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats
- Query
- Replication
- Schema Browser

### /select

class: org.apache.solr.handler.component.SearchHandler  
version: 4.10.2  
description: Search using components:

query  
facet  
mlt  
highlight  
stats  
expand  
clustering  
spellcheck  
debug

src: null

stats: handlerStart: 1417913145032  
requests: 19  
errors: 0  
timeouts: 0  
totalTime: 13519.270423  
avgRequestsPerSecond: 0.02771595355912348  
5minRateReqsPerSecond: 0.05413381234416733  
15minRateReqsPerSecond: 0.019781982315900518  
avgTimePerRequest: 711.5405485789474  
medianRequestTime: 136.1058  
75thPcRequestTime: 327.643716  
95thPcRequestTime: 9619.093724  
99thPcRequestTime: 9619.093724  
999thPcRequestTime: 9619.093724

## **v) Features that we implemented but did not figure into final deliverable**

1. Data Import Handler for Indexing news articles
  - a. We configured DIH and indexed the NYTimes corpus completely, but due to lack of fresh and relevant content we did not go ahead and use these pages as the seed pages to crawl
2. Parser in Java to parse Reuters Collection (RCV)
  - a. Parser was written in Java to get RCV articles into SOLR xml format
3. Seed Corpus from NYT or RCV
  - a. Both the seed corpora were not used as we found that a better way to generate fresh and breaking news is to use news aggregator sites such as [www.endmemo.com](http://www.endmemo.com) to get keywords for major world events over the last 5 years
4. AlchemyAPI for Keyword Extraction
  - a. We interacted with the AlchemyLanguage API to pull out most important keywords from the NYTimes and RCV seed pages. This worked well, but we did not use it as the NYT and RCV content were not good enough
5. Stanford NER
  - a. As mentioned in the sections above, we have chosen to do away with the feature when we failed to extract value out of it during index time and when we realized having NER applied on the stored contents of the article that the results are along expected lines we could not have it incorporated as it was already too late.

## **vi) Future Work**

1. Extract publishing time of an article. In this project, we have converted all DateTime to yyyy-mm-dd format because of inconsistencies in datetime formats across various sources. We could have refined our chronological view given more time
2. We can use Nutch in a better way to extract specific html tags from the web pages which would help to throw out unnecessary content (ads , links to facebook, twitter etc. ) on a web news article and thus improve the ranking of the articles . This can be done by writing custom parse plugins in Nutch
3. We could have better filtered/analyzed the web page content so that irrelevant cluster labels do not show up in the Topical View.
4. In the chronological view, we could have split the top 20-30 results into clusters and then pick one from each cluster and sort the picked documents by date to get related articles which will cover a different stage of a news story.
5. Can use date boosting to show the documents in chronological view
6. Integrate Solr AutoSuggest feature with our web app.
7. Work on making modifications to the Stanford NER library to fetch and return data in exactly the way the application would need
8. Crawl more news sites to get variety in the returned results/ related news

## **vii) Member Contributions**

### **Ashish Gupta:**

- Explored Apache Nutch and SOLR and setup a system to smartly crawl any news site and index them directly into SOLR index from Nutch. Overcame lot of issues we faced while crawling and indexing which have already been mentioned above. Help was taken from Girish for some problems faced.
- Wrote ruby scripts for interacting with Alchemy API , TheGuardian API and integration of SOLR index with the web app.
- Topical Summarization View – Explored various Clustering Algorithms and finally came up with an optimal way to cluster the documents via Lingo Clustering.
- Web App – Developed the RubyOnRails Web app which includes Server Side scripting, client side scripting and UI Design. Help was taken from Rajaram for the Chronological Summarization View.
- SOLR – Explored various SOLR filters to filter out unwanted content from news articles via the analyzer chain, which were hampering our search results. Helped Girish in the SpellCheck component and the Highlighting component both at SOLR side and the web app side.
- Both Girish and I explored various sites for breaking news keywords and trends for particular years. We narrowed down to endmemo.com and devised a strategy to smartly crawl the website and extract keywords.
- Both Girish and I realized that the seed corpora provided did not have content suited to our project and took the decision to go with only endmemo/Guardian/Hindu combination for developing our corpus
- Documentation – in coordination with Girish and Rajaram
- Discussion with the team in continuously improving our systems in ways that we deemed fit

### **Girish Suri:**

- Explored Data Import Handler and used it to index the Nytimes news corpus provided to us in coordination with Rajaram
- Implemented Spell Check, MoreLikeThis and Highlighting. Read solr wiki and understood the various parameters. A deeper understanding of the parameters helped in tuning our results. Ashish and Rajaram did the implementation of these features into the web application
- Explored Apache Nutch and together with Ashish figured out ways to extract the published dates from metatags
- Both Ashish and I explored various sites for breaking news keywords and trends for particular years. We narrowed down to endmemo.com and devised a strategy to smartly crawl the website and extract keywords.

- Both Ashish and I realized that the seed corpora provided did not have content suited to our project and took the decision to go with only endmemo/Guardian/Hindu combination for developing our corpus
- Found relevant solr parameters for clustering, appropriate filters to use in our schema, strategy to extract date using inbuilt pattern replace filters from Solr and then came with an apt schema to be used in our project
- Suggested the mlt.InterestingTerms feature to find related news stories which was done in coordination with Rajaram
- Discussion with the team in continuously improving our systems in ways that we deemed fit

#### Rajaram Rabinanth:

- Explored Data Import Handler and used it to index the NYtimes news corpus provided to us in coordination with Girish
- Prior to using data import handler for indexing:
  - Identifies -- Tags of importance present in the seed corpus
  - Wrote a simple java code to parse the given files to Solr input style doc xml files
- Implemented "More Like This" in Solr along with Ashish; also understood the different features in offer
- Crawled close to 7K documents on select topics from Guardian
- Worked on creating a regex to purge unwanted content from crawled pages belonging to the Guardian
- Spent significant amount of time on understanding tokenfilters and tokenizers; used knowledge gleaned to recommend components for our analyzer chain
- Spent a good amount of time on schema.xml; looked at sample schema files and experimented with our dummy setup to understand dos and don'ts, like define custom field types
- Recommended the usage of termVector; wrote a simple java program to query index and return term vectors of the documents returned
- Suggested test cases for the system and recommended shingles for multi-term query
- Stanford NER integration with the Solr system; followed the tutorial posted by SearchBox and helped integrate NER
- Discussion with the team in continuously improving our systems in ways that we deemed fit
- Webapp -- Worked on the base code of UI, provided by ashish to get the chronological view working; wrote a few lines of ruby code to accomplish (took Ashish's help in this regard)
- Documentation along with Girish and Ashish

Sai Srinath Sundar:

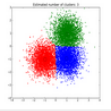
- Looked into topic modeling with the feasibility of implementing Latent Semantic Indexing and Latent Dirichlet Analysis analyzing the advantage of LDA over LSI.
- Explored Alchemy API (AlchemyLanguage) with focus on the keyword extraction and named entity recognition. Kept into consideration the API call limits which needed to be obeyed for extracting the same.
- Analyzed the carrot topic clustering features which include k-means clustering, suffix tree clustering and lingo clustering and helped in deciding which method to implement for the project.
- Explored Apache Solr's plethora of features particularly in indexing and tweaked a set of factory filters for optimal topic view results.
- Helped in the date extraction and in the removal of non-english languages from the crawled content.

## viii) Screenshots

### Home Page

# WARGS

Multi Dimensional News Summarization



Topics Related to sachin tendulkar

Events Related to sachin tendulkar

Showing Results for : sachin tendulkar

Total Search Results : 156

Sachin Tendulkar launches his autobiography – video | Sport | The Guardian

<http://www.theguardian.com/sport/video/2014/nov/06/sachin-tendulkar-launches-autobiography-playing-my-way-video>

Tendulkar looks back over his career, including winning the 2011 World Cup for India – 'It doesn't get any better than that in cricket' and the dressing room was 'flowing with champagne' – during the launch of his autobiography, *Playing It My Way*, in Mumbai on Wednesday. The book has already stirred controversy for its comments about the team's former coach Greg Chappell, who **Tendulkar** says targeted senior players unfairly Source: Reuters Thursday 6 November 2014 09:29 EST Share on Facebook Share on Twitter Share via Email Share on LinkedIn Share on Google+ Share on WhatsApp Topics **Sachin Tendulkar**

Sachin Tendulkar bats for clean water with Livpure - The Hindu

<http://www.thehindu.com/sport/sachin-tendulkar-bats-for-clean-water-with-livpure/article6244855.ece>

royals may file defamation suit against Chetan Bhagat Know your English — October 28, 2014 Reviews | Authors | Columns | Literary Review | I Know Your English | Children | Data Trending Videos Cricket Football Hockey Tennis Books Other Sports Blog Sport July

### Topical View

# WARGS

Multi Dimensional News Summarization



Topics Related to box office results for interstellar

Events Related to box office results for interstellar

Showing Results for : box office results for interstellar

Total Search Results : 17952

Interstellar Voyage through the Oculus Rift Wormhole

Widest-ever IMAX Rollout Anticipated Film the Guardian

Delivers Film the Guardian

Matthew McConaughey and the Cast of Interstellar

Total

Hopes

Catherine Shoard and Henry Barnes

Interstellar and Sci-fi's Obsession with Americana Popular

Space Odyssey

Other Topics

Notes on an Interstellar voyage through the Oculus Rift wormhole | Film | The Guardian

but Big Hero 6 wins in US Cosmos-hopping space opera takes \$132m across the world, but is six million shy of Disney's \$56m take in America Published: 10 Nov 2014 **Interstellar** dominates global box office but Big Hero 6 wins in US The Guardian Film Show: **Interstellar**, Leviathan, Say When and The Possibilities are Endless - video reviews The film team rocket through the wormholes and plot holes of **Interstellar** and raise a glass to hard-hitting, heavy-drinking Leviathan Published: 7 Nov 2014 The Guardian Film Show: **Interstellar**



## Chronological View

Topics Related to box office results for interstellar

Events Related to box office results for interstellar

Showing Results for : [box office results for interstellar](#)

Total Search Results : 17952

[Interstellar dominates global box office but Big Hero 6 wins in US | Film | The Guardian](#)

1 month ago

November 10, 2014

premiere in Washington DC. Photograph: Paul Morigi/WireImage Catherine Shoard Monday 10 November 2014 02:32 EST Share on Facebook Share on Twitter Share via Email Share on LinkedIn Share on Google+ Share on WhatsApp The canvas for Christopher Nolan's three-hour intergalactic drama stretches to distant galaxies accessible only through wormholes. So it's fitting that the film comprehensively dominated the global **box office** on its weekend of release, with a \$132m haul. Along with the \$50m scored in the US, the film took \$14m in Korea, \$8m in the UK and \$8m in Russia. The film has not yet opened in Japan and China, which should boost sales on the third week of release. But in America, Disney's latest, Big Hero 6, won the battle for **box office**

<http://www.theguardian.com/film/2014/nov/10/interstellar-box-office-big-hero-6> | [+Related news suggestions](#)

Matthew McConaughey and the cast of Interstellar: 'Mother nature's going to be just fine, it's us th  
Hollywood pins hopes on Interstellar as it seeks out new life in movie industry | Film | The Guardia

[Interstellar review – if it's spectacle you want, this delivers | Film | The Guardian](#)

1 month ago

November 09, 2014

Christopher Nolan Science fiction and fantasy Matthew McConaughey Jessica Chastain More... Anne Hathaway Share on Facebook Share on Twitter Share via Email Share on LinkedIn Share on Google+ Share on WhatsApp View all comments > comments Sign in or create your Guardian account to join the discussion. This discussion is closed for comments. We're doing some maintenance right now. You can still read comments, but please come back later to add your own. Commenting has been disabled for this account ( why? ) Order by newest oldest Show 25 25 50 100 Threads collapsed expanded unthreaded Loading comments... Trouble loading? View more comments more on this story **Interstellar dominates global box office but Big Hero 6 wins in US** Cosmos-hopping space opera takes \$132m across the world, but is six million shy of Disney's \$56m take in America Published: 10 Nov 2014 **Interstellar dominates global box office but Big Hero 6 wins in US** The Guardian Film Show: Interstellar, Leviathan, Say When and The Possibilities are Endless - video reviews The film team rocket through the wormholes and plot holes of **Interstellar**