CS 5604 Information Storage and Retrieval

Presenters: Andrej Galad, Long Xia, Shivam Maharshi, Tingting Jiang

Spring 2016 CS 5604

Information Retrieval and Storage

Virginia Polytechnic Institute and State University

Blacksburg, VA

Professor: Dr. E. Fox

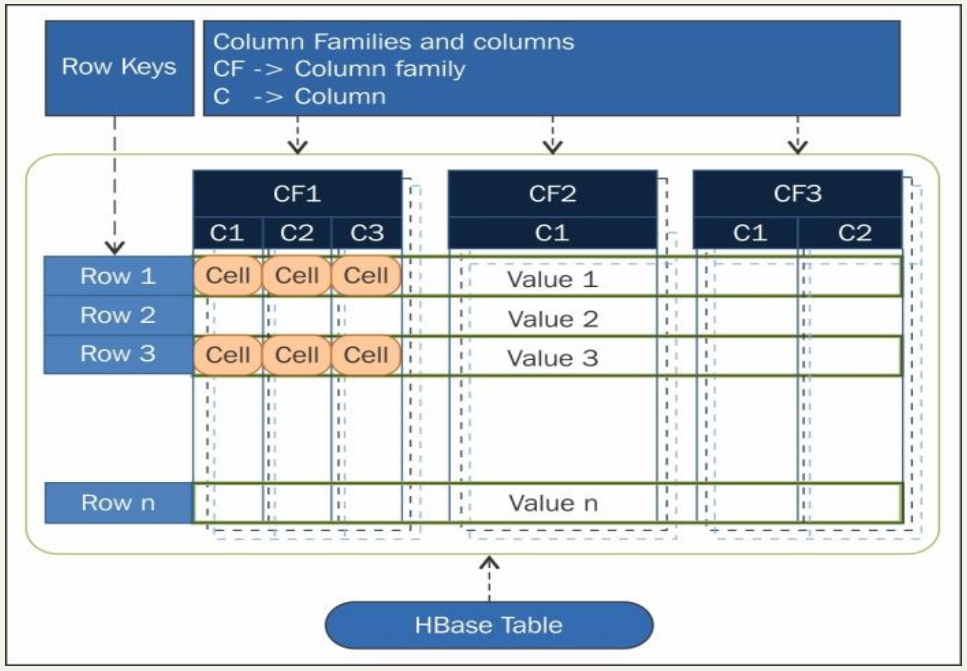
Project Overview

- Integrated Digital Event Archive and Library (IDEAL) project
  - Data source: social media (tweets, related web pages)
  - Goal: build a state-of-the-art information retrieval system
  - Management: separate teams, Solr team, Front-end team

- Solr team's responsibility
  - Data storage and HBase schema
  - Indexing
  - Custom search (query handler, ranking function, etc.)
  - Support for other teams (Front-end, Collaborative filtering)

Data Storage and HBase Schema

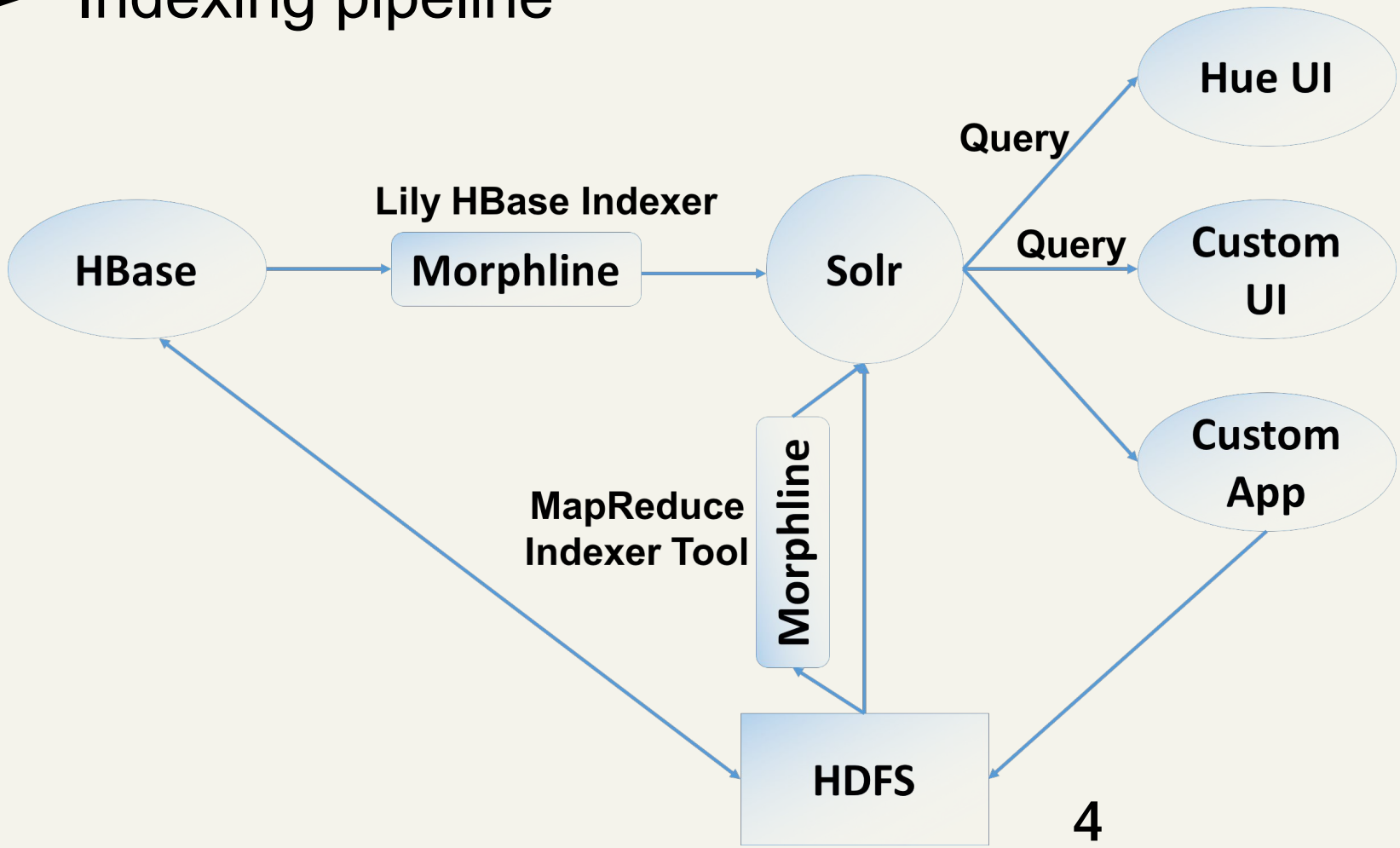
- Why use HBase
  - Non-relational, column-family-oriented, key-value-based database
  - Great scalability and flexibility
- How data stored



- HBase schema
- Import data into HBase

Indexing

- Indexing pipeline



Indexing

- Two types indexers
  - Lily HBase Batch Indexer
  - Lily HBase Near Real-time (NRT) Indexer
- Morphlines
  - Data extracting, transforming, and loading to Solr
  - Morphlines configuration file
- Solr Schema

Solr schema.xml & solrconfig.xml

- Static & Dynamic Fields

```
<schema name="ideal-cs5604s16" version="1.5">
 <fields>
   <field name="id" type="string" indexed="true"
stored="true" required="true" multiValued="false"
 />
   <field name="text" type="text_general" indexed
="true" stored="false" multiValued="true"/>
   <field name="_version_" type="long" indexed="t
rue" stored="true"/>
   <dynamicField name="*_i"  type="int"    indexe
d="true"  stored="true"/>
   <dynamicField name="*_is" type="int"    indexe
d="true"  stored="true"  multiValued="true"/>
   <dynamicField name="*_s"  type="string"  index
ed="true"  stored="true" />
   <dynamicField name="*_ss" type="string"  index
ed="true"  stored="true" multiValued="true"/>
   <dynamicField name="*_l"  type="long"   indexe
d="true"  stored="true"/>
   <dynamicField name="*_ls" type="long"   indexe
d="true"  stored="true"  multiValued="true"/>
   <dynamicField name="*_t"  type="text_general"
   indexed="true"  stored="true"/>
```

- Default & Copy Fields

```
<requestHandler name="/select" class="solr.SearchHandler">
   <lst name="defaults">
     <str name="echoParams">explicit</str>
     <int name="rows">10</int>
     <str name="df">text</str>
   </lst>
<copyField source="clean_text_s" dest="text"/>
<copyField source="clean_text_t" dest="text"/>
<copyField source="hashtags_s" dest="text"/>
<copyField source="urls_s" dest="text"/>
<copyField source="mentions_s" dest="text"/>
<copyField source="collection_name_s" dest="text"/>
<copyField source="domain_s" dest="text"/>
<copyField source="title_s" dest="text"/>
<copyField source="source_s" dest="text"/>
<copyField source="topic_label_ss" dest="text"/>
<copyField source="cluster_label_s" dest="text"/>
```

- Stop & Profanity words

```
    <fieldType name="text_general" class="solr.TextField"
positionIncrementGap="100">
      <analyzer type="index">
        <tokenizer class="solr.StandardTokenizerFactory"/>
        <filter class="solr.StopFilterFactory" ignoreCase=
"true" words="stoplist.txt" />
        <filter class="solr.StopFilterFactory" ignoreCase=
"true" words="profanity.txt" />
```

Morphline Configuration

- Mappings from Hbase cells to Solr fields (31 fields)
- Split fields into Multi-valued fields (4 fields)

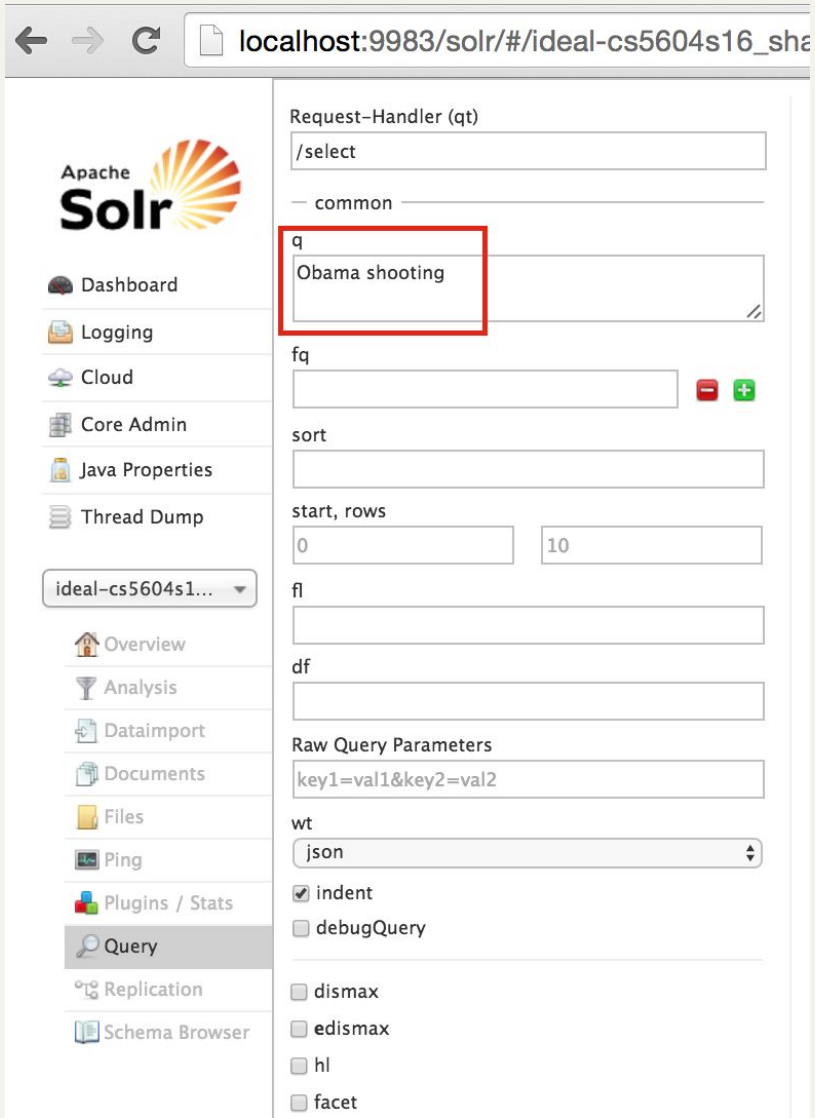
```
commands : [
  {
    extractHBaseCells {
      mappings : [
        {
          inputColumn : "clean_tweet:clean_text"
          outputField : "clean_text_s"
          type : string
          source : value
        }
```

```
{
  inputColumn : "tweet:geo_coordinates_0"
  outputField : "latitude_f"
  type : string
  source : value
}
```

```
{
  inputColumn : "cf_cf:sim_scores"
  outputField : "recommendation_sim_scores_s"
  type : string
  source : value
}
```

```
split
  {
    inputField : "recommendation_sim_scores_s"
    outputField : "recommendation_sim_scores_fs"
    separator : ";"
    isRegex : false
    addEmptyStrings : false
    trim : true
  }
```

Solr Search Admin UI



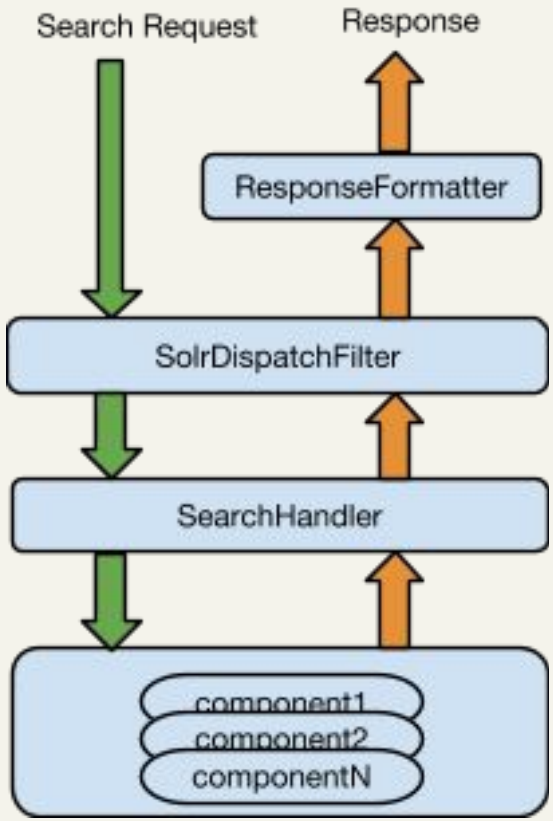
Custom Ranking

- Solr score (tf-idf) + custom scores (other teams)
  - $Custom\ Relevance\ Score = W_{Topic} * (Document\ Score)_{Topic} + W_{Clustering} * (Document\ Score)_{Clustering} + W_{Collection} * (Document\ Score)_{Collection}$
- Weight techniques
  - Multiple linear regressions
  - Empirical analysis for the Fractional Relevant Documents
- Ultimately…
  - Query Boosting + Query expansion + Re-ranking + Pseudo-Relevance feedback

Solr Search Components

- Solr - pluggable web application
  - Custom handlers, components, libraries
  - Dynamic linking
    - Custom classloaders
    - Declarative discovery - solrconfig.xml
    - Pain while debugging!!!
- Sample Component

```java
public class SampleComponent extends SearchComponent {
    @Override
    public void init(NamedList args) {
        // INITIALIZATION - invoked once ("constructor")
    }

    @Override
    public void prepare(ResponseBuilder rb) throws IOException {
        // PRE-PROCESSING - invoked before query is executed
    }

    @Override
    public void process(ResponseBuilder rb) throws IOException {
        // POST-PROCESSING - invoked after all the results are fetched
    }

    @Override
    public String getDescription() {
        return "Sample description";
    }

    @Override
    public String getSource() {
        return "Some repository";
    }
```



Solr: Component Configuration

1. Build and upload JAR(s) + dependencies to all Solr nodes

```
[cs5604s16_so@solr2 ideal]$ pwd
/home/cs5604s16_so/bin/ideal
[cs5604s16_so@solr2 ideal]$ ll -a
total 8436
drwxrwxrwx 2 cs5604s16_so cs5604s16_so    4096 Apr 24 22:21 .
drwxrwxrwx 4 cs5604s16_so cs5604s16_so    4096 Apr  9 11:57 ..
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so   20756 Apr 24 22:05 hbase-annotations-1.0.0.jar
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so 1095441 Apr 24 22:05 hbase-client-1.0.0.jar
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so  507776 Apr 24 22:05 hbase-common-1.0.0.jar
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so 3689063 Apr 24 22:05 hbase-protocol-1.0.0.jar
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so 1475955 Apr 24 22:05 htrace-core-3.1.0-incubating.jar
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so   44095 Apr 24 22:05 IDEAL-1.0.jar
-rwxrwxrwx 1 cs5604s16_so cs5604s16_so 1779991 Apr 24 22:05 netty-all-4.0.23.Final.jar
-rw-r--r-- 1 cs5604s16_so cs5604s16_so      53 Apr 24 22:21 weights.conf
```

Solr: Component Configuration

1. Build and upload JAR(s) + dependencies to all Solr nodes
2. Register component in solrconfig.xml

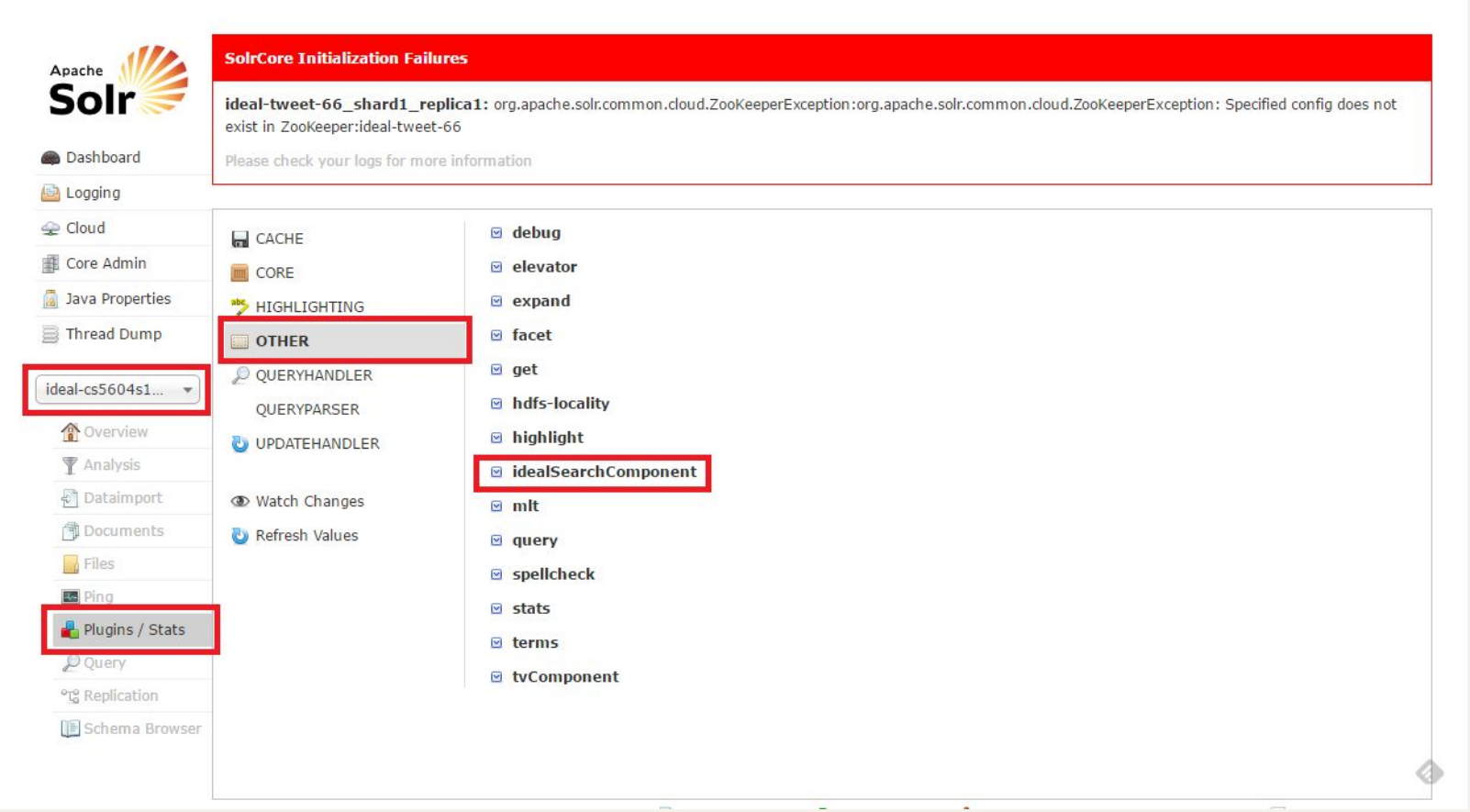
```
    When a 'regex'     <!-- If the default list of SearchComponents is not desired, that
    files in that c         list can either be overridden completely, or components can be
    (anchored on bc         prepended or appended to the default list.  (see below)
                          -->
    If a 'dir' opti
    is found that m       <arr name="last-components">
                            <str>idealRankingComponent</str>
    The examples be       </arr>
    with their exte
  -->                     </requestHandler>
<lib dir="../../../c
<lib dir="../../../c    <searchComponent name="idealRankingComponent" class="edu.vt.ideal.IDEALRankingComponent">
                          <str name="verbose">false</str>
<lib dir="../../../c      <str name="weights-file">/home/cs5604s16_so/bin/ideal/weights.conf</str>
<lib dir="../../../c    </searchComponent>

<lib dir="../../../c
<lib dir="../../../dist/" regex="solr-langid-\d.*\.jar" />

<lib dir="/home/cs5604s16_so/bin/velocity/contrib/velocity/lib" regex=".*\.jar" />
<lib dir="/home/cs5604s16_so/bin/velocity/" regex="solr-velocity-\d.*\.jar" />

<lib dir="/home/cs5604s16_so/bin/ideal/" regex=".*\.jar" />

<!-- an exact 'path' can be used instead of a 'dir' to specify a
     specific jar file.  This will cause a serious error to be logged
     if it can't be loaded.
```

Solr: Component Configuration

1. Build and upload JAR(s) + dependencies to all Solr nodes
2. Register component in solrconfig.xml
3. Update configuration and reload collection
   - $ solrctl instancedir --update <collection_name> <collection_configuration>
   - $ solrctl collection --reload <collection_name>

```
[cs5604s16_so@node1 ~]$ solrctl instancedir --update ideal-cs5604s16 ideal-cs5604s16/
Uploading configs from ideal-cs5604s16//conf to solr2.dlrl:2181,node2.dlrl:2181,node3.dlrl:2181,node1.dlrl:2181,node4.dl
rl:2181/solr. This may take up to a minute.
[cs5604s16_so@node1 ~]$ solrctl collection --reload ideal-cs5604s16
```

Solr: Component Verification



Query Manipulation

- Query Expansion
  - In-memory Lucene index based on *ideal-cs5604s16-topic-words*
  - Schema: label, collection_id, words

```
Set<String> searchTopicLabels(Collection<Term> terms) throws IOException {
    // only if we have an index
    if (searcher == null)
        return null;

    // creating synthetic query
    BooleanQuery query = new BooleanQuery();
    for (Term term : terms) {
        // if query contains explicit collection number narrow the search down to only relevant topics
        if (term.field().equals("colnum_s"))
            query.add(new TermQuery(new Term(COLLECTION_FIELD, term.text())), BooleanClause.Occur.MUST);
        else // else use the text to search in words
            query.add(new TermQuery(new Term(WORDS_FIELD, term.text())), BooleanClause.Occur.SHOULD);
        query.setMinimumNumberShouldMatch(1);
    }
    if (this.verboseMode)
        logger.info(query);

    // getting results
    TopDocs topDocs = searcher.search(query, MAX_RESULTS);
    if (this.verboseMode)
        logger.info(String.format("Found [ %s ] matches", topDocs.totalHits));
    if (topDocs.totalHits > 0) {
        Set<String> labels = new HashSet<>();
        for (ScoreDoc scoreDoc : topDocs.scoreDocs) {
            IndexableField labelField = searcher.doc(scoreDoc.doc).getField(LABEL_FIELD);
            if (labelField != null && labelField.stringValue() != null)
                labels.add(labelField.stringValue());
        }
        if (this.verboseMode)
            logger.info(String.format("Match found for some of the terms. Topic labels %s", labels));
        return labels;
    }
```

Query Manipulation

- Re-ranking
  - Tf-idf + weight $_1$ * custom score $_1$ + ...

```
@Override
protected CustomScoreProvider getCustomScoreProvider(AtomicReaderContext context) throws IOException {
    return new CustomScoreProvider(context) {
        @Override
        public float customScore(int doc, float subQueryScore, float[] valSrcScores) throws IOException {
            // original score - tf-idf
            float score = super.customScore(doc, subQueryScore, valSrcScores);

            if (verboseMode)
                logger.info(String.format("DocId [ %s ], original score [ %s ]", doc, score));

            Document d = context.reader().document(doc);
            // boosts to the score based on normalized value of certain fields
            for (String field : fieldWeights.keySet()) {
                IndexableField scoreField = d.getField(field);
                if (scoreField != null && scoreField.numericValue() != null) {
                    float scoreBoost = scoreField.numericValue().floatValue();
                    float weight = fieldWeights.get(field);
                    if (verboseMode)
                        logger.info(String.format("DocId [ %s ], field [ %s ], score boost [ %s ], weight [ %s ]"
                                doc, field, scoreBoost, weight));
                    score += weight * scoreBoost;
                }
            }

            if (verboseMode)
                logger.info(String.format("DocId [ %s ], final score [ %s ]", doc, score));

            return score;
        }
    };
}
```

Pseudo Relevance Feedback

```
@Override
public void process(ResponseBuilder rb) throws IOException {
  if (verboseMode)
    logger.info("IDEAL Ranking Component prepare phase invoked.");

  DocListAndSet searchResults = rb.getResults();
  DocList resultDocList = searchResults.docList;
  DocIterator docIt = resultDocList.iterator();
  int count = 0;
  // Fetch top-k results from the expanded query.
  List<Document> topKRes = new ArrayList<Document>();
  while (docIt.hasNext() && count < PRF_TOP_K) {
    Document document = rb.req.getSearcher().doc(docIt.nextDoc());
    if (document != null) {
      topKRes.add(document);
    }
    count++;
  }
```

Pseudo Relevance Feedback

```
  // Create a new query by using the fields from these top-k results.
  Query expandedQuery = rb.getQuery();
  BooleanQuery prfQuery = new BooleanQuery();
  for (Document doc : topKRes) {
    // Take relevance from collection field.
    TermQuery collectionQuery = null;
    String collectionField = doc.get(COLLECTION_FIELD);
    String collectionProb = doc.get(COLLECTION_PROB_FIELD);
    if (!Strings.isNullOrEmpty(collectionField) && !Strings.isNullOrEmpty(collectionProb)) {
      collectionQuery = new TermQuery(new Term("text", collectionField));
      collectionQuery.setBoost(expandedQuery.getBoost() * ((Float.valueOf(collectionProb))));
      prfQuery.add(collectionQuery, BooleanClause.Occur.SHOULD);
    }
  // Custom scores = tf-idf*1 + topics * tWeight + cluster *cWeight...
  rb.setQuery(new ScoreBoostingQuery(prfQuery, fieldWeights, verboseMode));
}
```

Problems Faced

- Reflection, reflection, reflection
- Lack of solid documentation
  - attempt => failure
- Cluster upgrade
  - API versions mismatch
- Getting the data into HBase from all teams
- Pain to debug Solr on cluster
- Insufficient access privilege to the cluster

Lessons Learned

- Clear scope and requirement details
- Clear contract and team deliverables
- More effective communications

Future Work

1. Precision and Recall evaluation
2. Performance improvements
3. Calculate the weights for custom ranking

Acknowledgement

- NSF grant IIS - 1319578, III: Small: Integrated Digital Event Archiving and Library (IDEAL)
- Dr. Edward A. Fox
- GRA: Sunshin Lee and Mohamed Magdy Farag
- All other teams