

IBM Watson

# Retrieve and Rank

Watson Services Deep Dive

IBM



# Overview

## What is it?

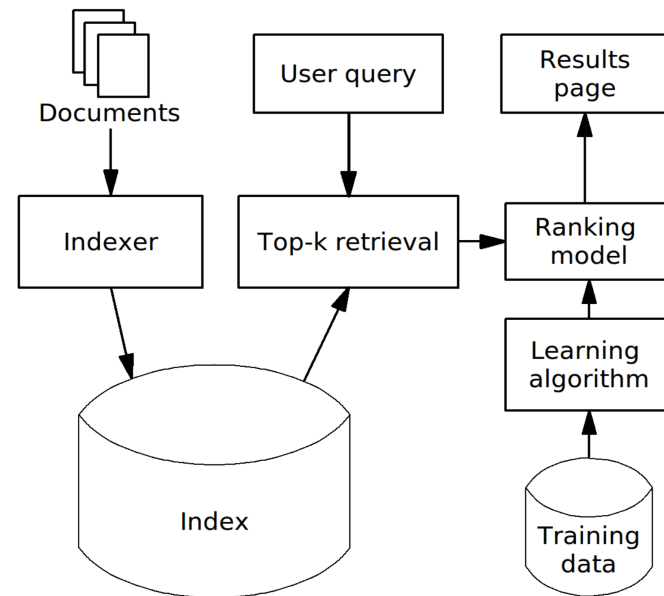
- Find highly relevant results from pre-loaded customer data using a trained ranking model built on Ground Truth
  - Will re-ranks lists of values based on a trained model
  - Fills need to index and search content scalably in cloud.

## Primary Use Cases

- Use cases where customer is seeking to enable easy retrieval of information from large quantities of information (Regulatory, Manufacturing, Retail, R & D, etc.)
  - Contact centers, field technician support pages, etc

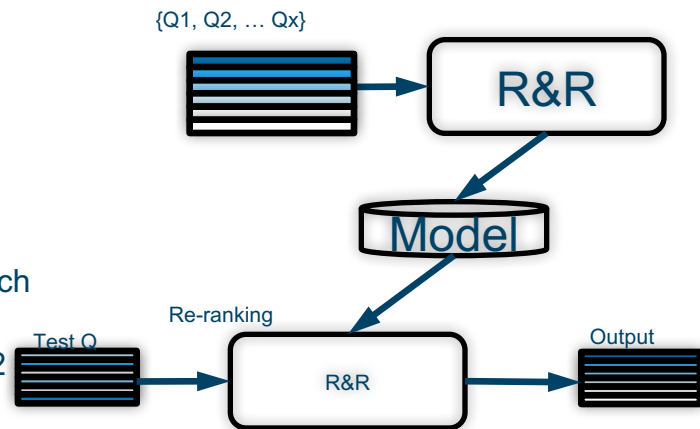
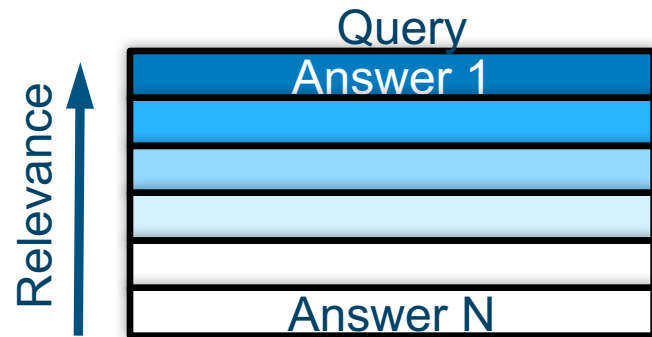
## Value Proposition

- Improved responses in comparison to standard search techniques by finding the “signal” in the data through machine learning



# How It Works

- Retrieve and Rank service combines two information retrieval components in a single service:
  - Search using Apache Solr (Retrieve)
  - Sophisticated machine learning capability to provide improved results (Rank)
    - Train a machine learning model based on known relevant results
    - Detect "signals" in the data.
  - This combination provides users with more relevant results by automatically re-ranking by Ranker.
- R&R learns to reorder (previously unseen) list of items based on model:
  - Input items belonging to a **list** (e.g. answers to question)
  - Each item represented by its **feature vector** for that list (e.g. match scores, counts, ...)
  - Each feature vector is also labeled with **ground truth** (e.g. 0, 1, 2 ...)



# Example

## Simple Support Center Example

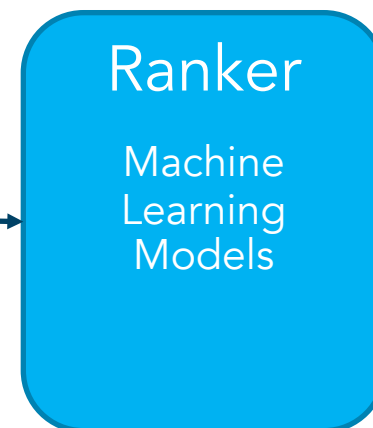
Solr contains 3 key fields: Short Description, Long Description, and Tech Notes

"What does the X221 error message mean?"



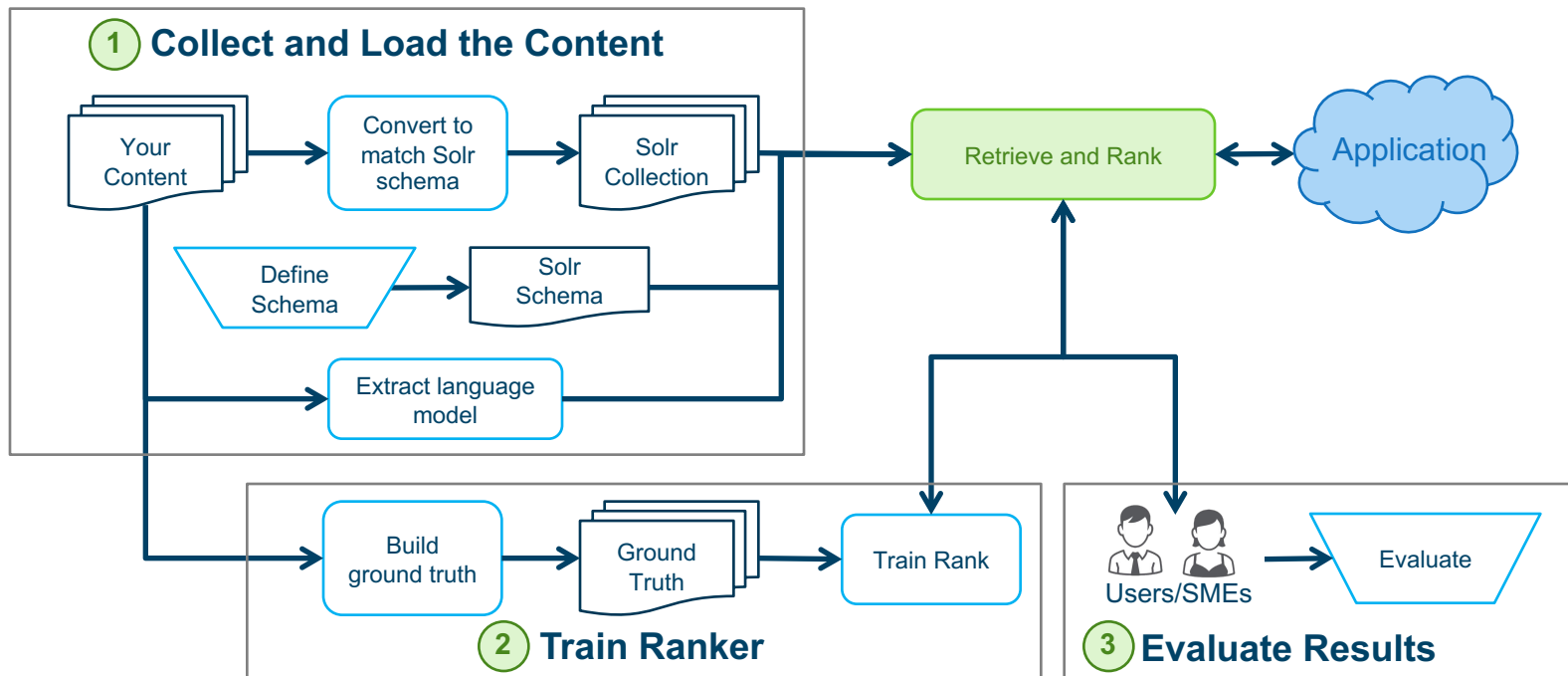
|               | Field 1 - Short Description |                 |                 | Field 2 - Long Description |                 |                 | Field 3 - Tech Note |                 |                 |
|---------------|-----------------------------|-----------------|-----------------|----------------------------|-----------------|-----------------|---------------------|-----------------|-----------------|
|               | Feature Score 1             | Feature Score 2 | Feature Score 3 | Feature Score 1            | Feature Score 2 | Feature Score 3 | Feature Score 1     | Feature Score 2 | Feature Score 3 |
| Answer ID #1  | 0.53                        | 0.10            | 0.61            | 0.70                       | 0.40            | 0.15            | 0.82                | 0.09            | 0.32            |
| Answer ID #2  | 0.45                        | 0.02            | 0.34            | 0.57                       | 0.39            | 0.83            | 0.22                | 0.94            | 0.21            |
| Answer ID #3  | 0.95                        | 0.35            | 0.86            | 0.82                       | 0.02            | 0.23            | 0.13                | 0.01            | 0.52            |
| Answer ID #4  | 0.29                        | 0.02            | 0.68            | 0.21                       | 0.83            | 0.19            | 0.68                | 0.27            | 0.15            |
| Answer ID #5  | 0.16                        | 0.62            | 0.11            | 0.93                       | 0.06            | 0.93            | 0.81                | 0.68            | 0.89            |
| Answer ID #6  | 0.24                        | 0.06            | 0.69            | 0.78                       | 0.31            | 0.34            | 0.70                | 0.03            | 0.73            |
| Answer ID #7  | 0.29                        | 0.57            | 0.76            | 0.46                       | 0.71            | 0.61            | 0.29                | 0.34            | 0.53            |
| Answer ID #8  | 0.81                        | 0.87            | 0.07            | 0.14                       | 0.45            | 0.11            | 0.74                | 0.25            | 0.37            |
| Answer ID #9  | 0.79                        | 0.24            | 0.44            | 0.13                       | 0.29            | 0.52            | 0.82                | 0.79            | 0.51            |
| Answer ID #10 | 0.42                        | 0.60            | 0.59            | 0.96                       | 0.64            | 0.77            | 0.90                | 0.36            | 0.72            |
| .....         |                             |                 |                 |                            |                 |                 |                     |                 |                 |

|               |
|---------------|
| Answer ID #7  |
| Answer ID #2  |
| Answer ID #8  |
| Answer ID #4  |
| Answer ID #1  |
| Answer ID #25 |
| Answer ID #33 |
| Answer ID #18 |
| ...           |



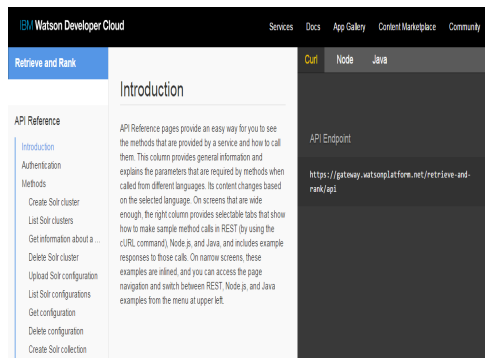
\*IBM Watson Solr Plugin contains a set of 'scorers' that scores each data attribute (Feature Score)

# R&R Workflow - Overview



# Workflow Tasks

- Getting started with the Retrieve and Rank service
  - [https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/retrieve-rank/get\\_start.shtml](https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/retrieve-rank/get_start.shtml)



## 1. Create Solr Cluster

Specify cluster name and size.  
Wait until the cluster is available.

## 2. Create and upload Solr Configuration

Define your schema and upload the packaged configuration files.

## 3. Create Solr Collection

Create Solr Collection associated with the Solr Configuration

## 4. Upload and index Documents

Gathering documents to be retrieved and upload them into Solr Collection.

## 5. Create/Update Ground Truth

Consolidate example questions and relevant answers with relevance scores.

## 6. Create and Train Ranker

Generate a training data from the relevance file and upload it to create Ranker

## 7. Evaluate result / update Ground Truth

Wait the training completion and try/evaluate the re-ranked results

*When you change your training data, you must create a new ranker with your modified data*

# Retrieve

# Solr - Overview

- Open source enterprise search platform, written in Java, from the Apache Lucene project.
- Its major features include
  - Full-text search
  - Hit highlighting
  - Faceted search
  - Real-time indexing
  - Dynamic clustering
  - Database integration
  - NoSQL features
  - Rich document (e.g., Word, PDF) handling
  - Providing distributed search and index replication
- Designed for scalability and Fault tolerance.
- The most popular enterprise search engine.

[https://en.wikipedia.org/wiki/Apache\\_Solr](https://en.wikipedia.org/wiki/Apache_Solr)

- Not all Solr features are available on Retrieve and Rank Service



# Solr - Configuration

- Solr configuration ZIP file contains schema and system configuration files
- Schema configuration file (schema.xml)
  - Describes the structure of your documents.
    - Which fields will be indexed and searched in documents
    - How the field contents values are tokenized and filtered.
- System configuration file (solrconfig.xml)
  - Defines IBM extensions of plug-ins for Solr
    - Query Parser
    - Feature Generator
- Blank configuration file for the Retrieve and Rank service are available at [Configuring the Retrieve and Rank service](#)
  - For details about how to configure Solr in general, search for schema design in the [Apache Solr reference](#).
- Be careful not to include root folder into the ZIP file !

```
[<root>]
| currency.xml
| protwords.txt
| schema.xml
| solrconfig.xml
| stopwords.txt
| synonyms.txt
└─[lang]
   stoptags_ja.txt
   stopwords_en.txt
   stopwords_ja.txt
```

# Solr - Schema

```
<field name="id" type="string" indexed="true" stored="true"
      required="true" multiValued="false" />
<field name="title" type="watson_text_en" indexed="true" stored="true"
      required="false" multiValued="true" />
<field name="author" type="watson_text_en" indexed="true" stored="true"
      required="false" multiValued="true" />
<field name="bibliography" type="watson_text_en" indexed="true" stored="true"
      required="false" multiValued="true" />
<field name="body" type="watson_text_en" indexed="true" stored="true"
      required="false" multiValued="true" />
```

## Field types

- String (solr.StrField)
  - A word/sentence stored as an exact string without performing tokenization etc. Commonly useful for storing exact matches, e.g, for facetting.
- Text (solr.TextField)
  - A sentence typically performed tokenization, and secondary processing (such as lower-casing etc.). Useful for all scenarios when we want to match part of a sentence.

```
<fieldType name="watson_text_en" indexed="true" stored="true"
class="com.ibm.watson.hector.plugins.fieldtype.WatsonTextField">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.EnglishPossessiveFilterFactory"/>
    <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
    <filter class="solr.PorterStemFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_en.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.EnglishPossessiveFilterFactory"/>
    <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
    <filter class="solr.PorterStemFilterFactory"/>
  </analyzer>
</fieldType>
```

# Solr - Config

- Defines default behavior of Solr.
  - Default search fields, handlers, etc.
- Not all base Solr features supported
  - Property enableRemoteStreaming set to true:
  - Any `<updateHandler></updateHandler>` elements.
  - Configs that do not include `<updateLog />`.
  - Overwritten updateLog dir:
    - The value must be `${solr.ulog.dir:}`.
  - Overwritten Solr data dir:
    - The value must be `${solr.data.dir:}`.
  - Any `<jmx ... />` elements.
  - Use of AdminHandlers:
  - Use of ReplicationHandler:

```
<queryParser name="fcQueryParser"
              class="com.ibm.watson.hector.plugins.qparser.WatsonFCQParser">
  <str name="textFieldTypes">watson_text_en</str>
</queryParser>
<searchComponent name="fcFeatureGenerator"
                  class="com.ibm.watson.hector.plugins.ss.FCFeatureGeneratorComponent" />

<!-- feature centric document search -->
<requestHandler name="/fcselect"
                class="com.ibm.watson.hector.plugins.ss.FCSearchHandler">
  <lst name="defaults">
    <str name="defType">fcQueryParser</str>
  </lst>
  <arr name="last-components">
    <str>fcFeatureGenerator</str>
  </arr>
</requestHandler>
```

# Solr - Collection Documents

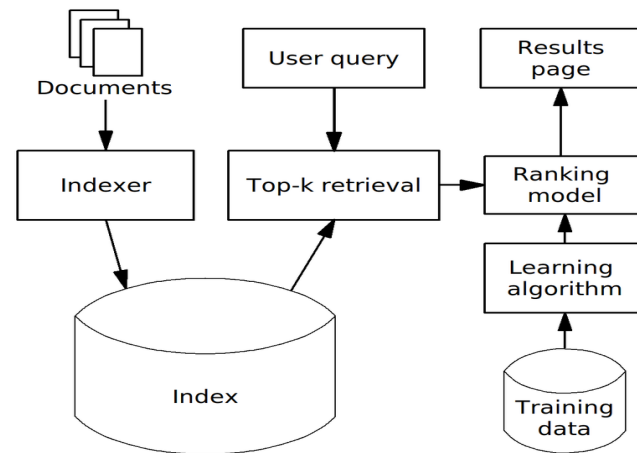
- Document consistent with fields defined in the Schema.
- Commit action
  - Or changes will not be visible to search.
- Multiple content type support
  - JSON, XML

```
{
  "add" : {
    "doc" : {
      "id" : 1,
      "author" : "brenckman,m.",
      "bibliography" : "j. ae. scs. 25, 1958, 324.",
      "title" : "experimental investigation of the aerodynamics of a wing in a slipstream .",
      "body" : "experimental investigation...",
    }
  },
  ...
},
"commit" : { }
```

# Rank

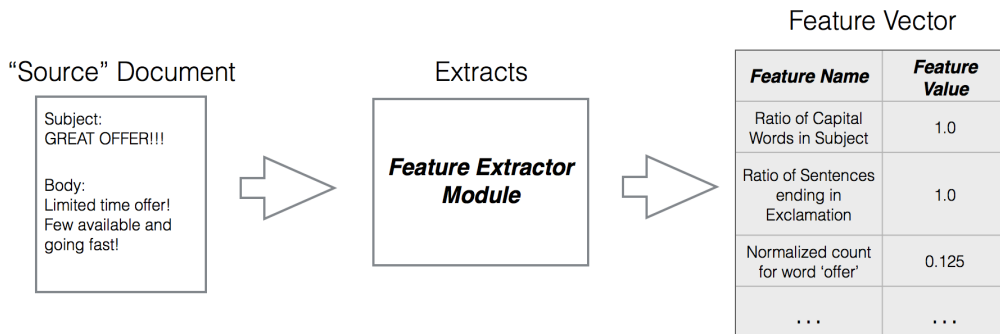
# Ranker - Overview

- Based on well-know research output – Learn to Rank
  - Machine learned ranking using supervised or reinforcement learning.
  - Training data consist of items with partial ordering using numerical “relevance” score
  - Ranker will apply an ordering to new/unseen lists in way that is similar to ranking in training data.
- Different from general classification
  - Optimization is list-wise
    - All list items considered when learning model parameters (rather than each item separately)
  - Allow for graded relevance
    - Multiple level (0, 1, ...) represent preferential ordering



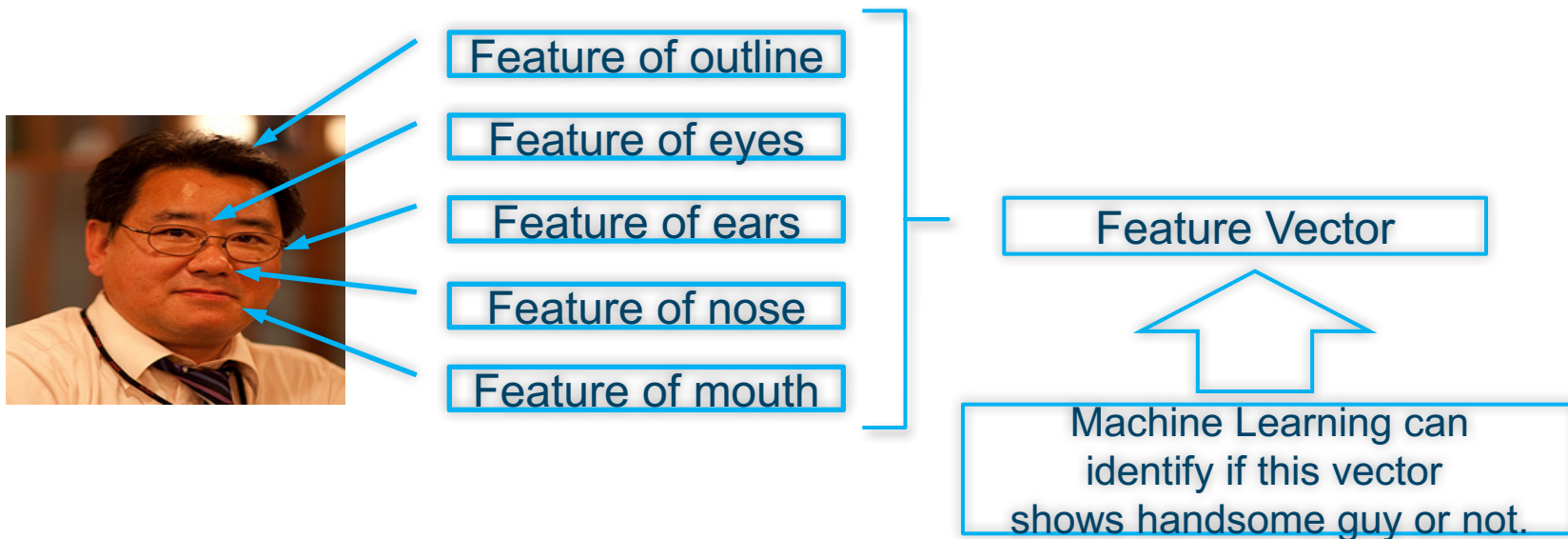
# Features / Feature Vectors

- Machine Learning performs a task (finds a pattern, makes a prediction)
  - Determine if email is spam or not.
- Feature Engineering
  - Finding the attributes/information/elements that are useful to make the task prediction
- Feature is a numerical representation of an object
- Scoring/Extraction
  - Finding the value of the feature



# Features / Feature Vector

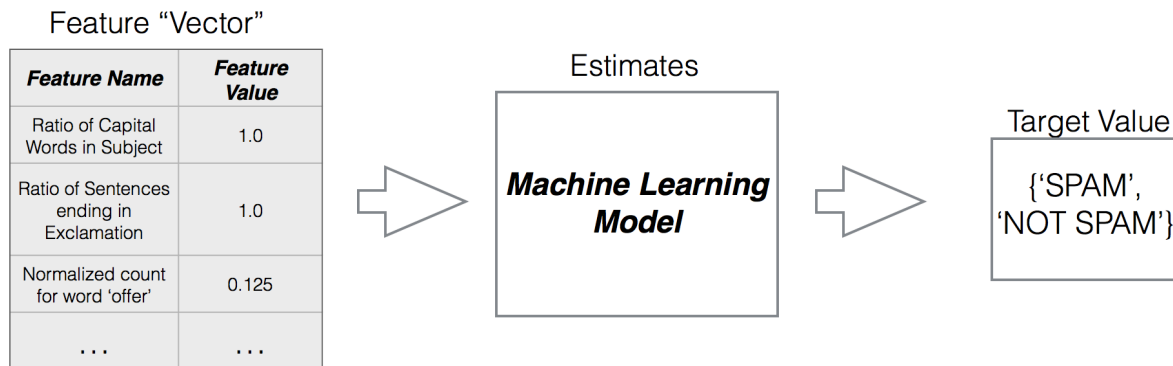
- Feature vector is combined feature values





# Model

- Machine learning model maps features to a value in the target output space.
- Machine learning algorithm selects the model that best fits the training data.



# Ranker - Features

- Features in Retrieve and Rank
  - Several text based key word search results (n-grams)
  - Generated by IBM Plug-ins on Solr (Retrieve)
  - Features are generated for both fields and document wide
  - Ranker see only feature vector and does not know anything about contents of the documents.
    - Language independent
- Create and train Ranker
  - Feed a collection of the feature vectors which were generated from the known questions and answers with relevant score.
  - Ranker understand weighting of features and select suitable ranking algorithm in this business domain

# Training Data - Artifacts

- **Ground truth**
  - Collection of questions that are matched to answers.
  - Each answer is labeled with their **relevance** to the question.
    - The relevance label (score) tell the ranker which features are the most useful.
- **Relevance file**
  - CSV file contains Ground Truth data.
- **Training Data**
  - CSV file contains questions ID, Feature Vector generated by the IBM Plug-in, and relevance score.
    - The relevance label tell the ranker which features are the most useful.
  - Generated by train.py script
  - A Ranker is create from the Training Data, not Ground Truth (nor Relevance file).

# Training Data - Relevance File (Ground Truth)

- Relevance file is CSV format representation of Ground Truth.
  - The relevance file meets the training data quality standards.
- Values:
  - {question} = Question phrase
    - This phrase is used as query for /fcselect and the query character restriction for is applied.
  - {answer\_idX} = "id" field value of a document in the index.
  - {relevance\_labelX} = Integer value to express the relevance of the answer.
    - 0 : irrelevant
    - 1 : least relevant
    - N+1 : More relevant than N

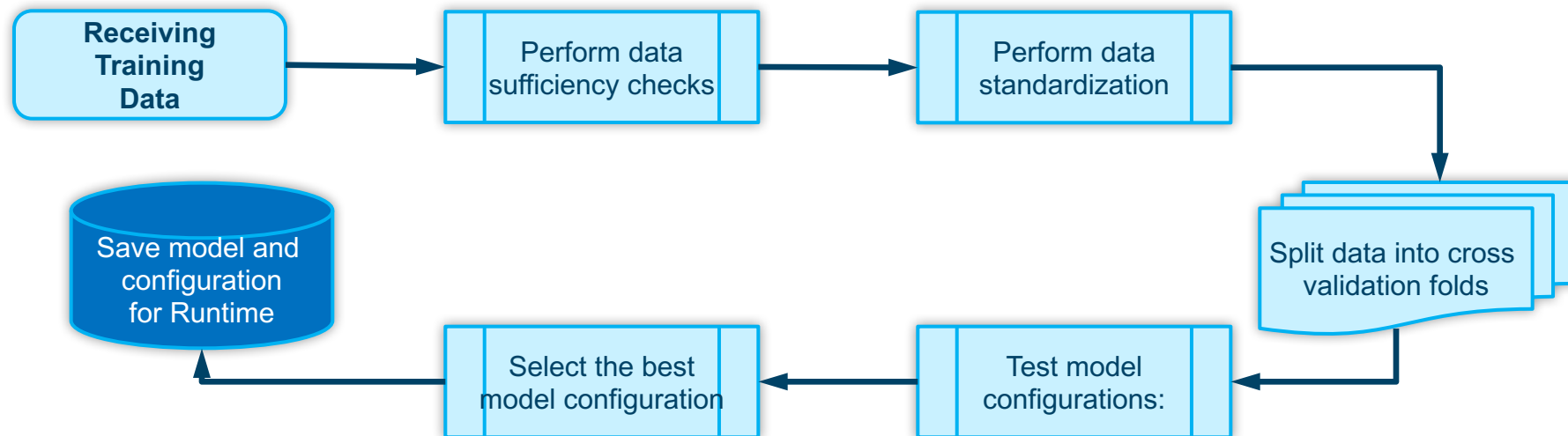
- "0" : an answer is "not relevant"
- "1" : "somewhat relevant"
- "2" : "relevant"
- "3" : "very relevant"
- "4" : a "perfect" answer

```
"{question}","{answer_id1}","{relevance_label1}","{answer_id2}","{relevance_label2}", ...
```

# Training Data - Requirements

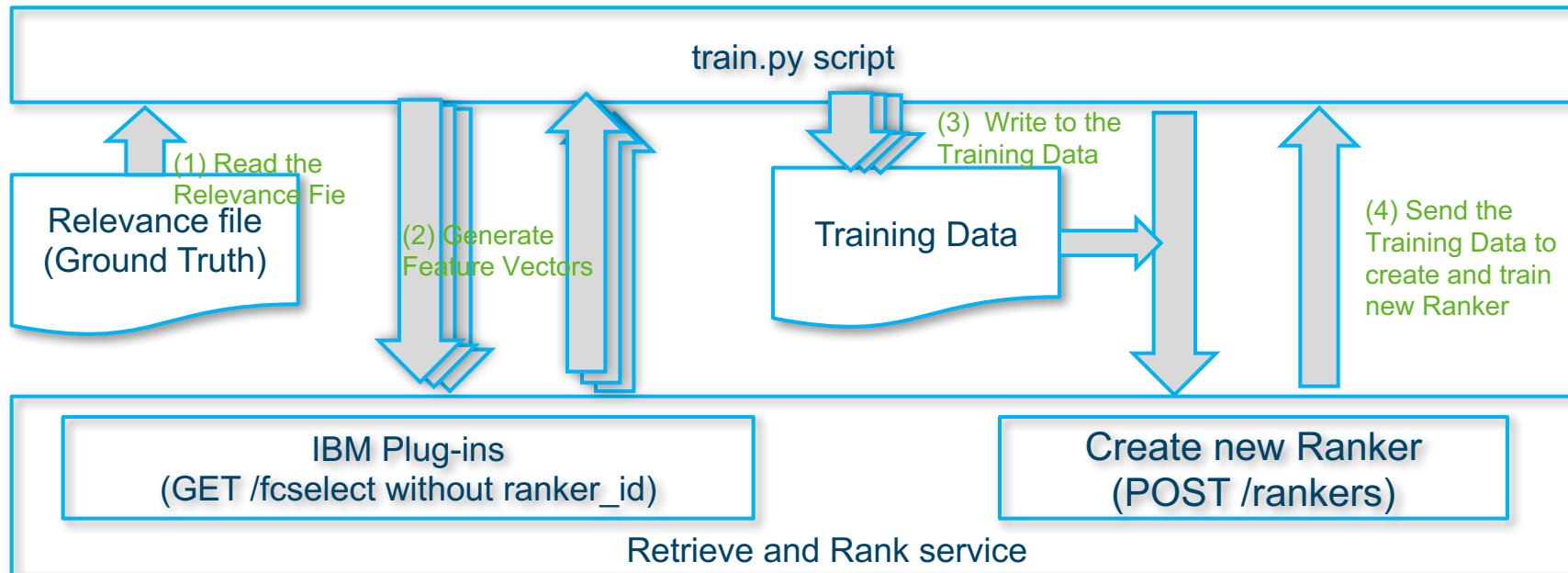
- At least 49 unique questions.
- The number of records (questions) must be at least 50 times the number of indexed fields
  - For example, 5 fields definition in `schema.xml` requires at least 250 records in your training data.
- At least two different relevance labels must exist in the data and those labels must be well represented.
  - “A well represented label” means It occurs at least once for every 100 unique questions.
    - For example, you have 300 unique questions and you use a relevance scale of 0,1,2,3. At least two distinct labels (for example, 0 and 2) must each appear three times ( $0.01 \times 300$ ) in the training data.

# Training Data - Ranker Creation Process



# Training Data - Creation Script

- [train.py](#) script helps you to create Training Data and upload it for creating and training new Ranker.



# Runtime Search - Reserved Characters

- The following characters have special usage in queries:
  - Ampersand (&)
    - Reserved by the HTML parser and used to separate parameters in the REST request.
  - Colon (:)
    - Escape a colon in a query with a backslash.
  - Double quotation marks (")
    - Escape double quotation marks in a query with a backslash in field queries.
  - Backslash (\)
    - Escape a backslash in a query with another backslash.



# Runtime Search - Re-Rank Results

- When the Ranker is “Available”, you can issue a questions as the following command.
  - The Retrieve and Rank service does Solr search and re-rank with this single request.

```
https://{username}:{password}@gateway.watsonplatform.net/retrieve-and-rank/api/  
v1/solr_clusters/{solr_cluster_id}/solr/example-collection/fcselect?ranker_id={ranker_id}&q={your question}&wt=json&fl=id,title
```

# Runtime Search - Separate Processes of R&R

- The Search of Retrieve and Rank service is done by IBM plugged-in Query Parser and the result is different with standard Solr search by "/select"
- To mimic the internal process of the Retrieve and Rank service, follows
  - Call "/fcselect" without ranker\_id, but with the following parameters to generate feature vector.
  - Save the value of <str name="RSInput"> of the response into CSV file (retrieved\_answers.csv).
  - Call ranker to re-rank the retrieved result in the CSV

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<lst name="responseHeader"><int name="status">0</int><int name="QTime">4</int></lst><result name="response" numFound="8">
<doc><str name="id">1427</str></doc><doc><str name="id">872</str></doc><doc><str name="id">878</str></doc><doc><str name="id">945</str></doc><doc><str name="id">881</str></doc><doc><str name="id">875</str></doc><doc><str name="id">876</str></doc></result><str name="RSInput">answer_id,f0,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,r1,r2,s
879,4.68243,2.7150407,1.1170104,2.8128014,1.0975189,0.5167495,0.19321924,0.5167495,3.814578,1.9315135,0.70725864,2.08
1427,4.651482,2.7183692,1.1457773,2.7183692,0.91731715,0.4040944,0.19321924,0.4040944,3.924729,2.0695136,0.70725864,2
872,4.6083317,2.6608353,1.1170104,2.7343855,0.0,0.0,0.0,0.0,0.4.863838,2.8690553,1.236312,2.9451354,0.5,2,0.28768207245
874,4.3044786,2.2710023,1.070195,2.2710023,1.0889331,0.48698258,0.18749675,0.48698258,3.2404234,1.3403438,0.63167626,
878,4.1892443,2.5924177,1.1994255,2.8434803,1.0889331,0.5053045,0.18749675,0.5053045,2.99195,1.6678681,0.7732144,1.66
945,3.6964252,1.908088,0.6277686,1.908088,0.5523116,0.19993505,0.0,0.0,0.19993505,3.353797,1.7040727,0.70725864,1.7040727
881,3.6577005,1.5900877,0.46876836,1.5900877,0.0,0.0,0.0,0.0,0.3.890311,1.7515132,0.5482584,1.7515132,0.5,8,0.133531392
875,3.6532748,1.850554,0.5990015,1.9359043,0.0,0.0,0.0,0.0,0.3.8815784,2.0119796,0.6784916,2.100745,0.5,7,0.11778303565
63,3.4579248,1.5900877,0.46876836,1.5900877,0.0,0.0,0.0,0.0,0.3.8113556,1.9183488,0.63167626,1.9183488,0.5,8,0.10536051
876,3.721106,2.1551426,1.1871233,2.1551426,0.91731715,0.4040944,0.19321924,0.4040944,2.6272507,1.3594499,0.76090676,1
</str>
</response>
```

[https://{username}:{password}@gateway.watsonplatform.net/retrieve-and-rank/api/v1/solr\\_clusters/{solr\\_cluster\\_id}/solr/kyotoCollection/fcselect?q={your question}&returnRSInput=true&generateHeader=true&fl=id](https://{username}:{password}@gateway.watsonplatform.net/retrieve-and-rank/api/v1/solr_clusters/{solr_cluster_id}/solr/kyotoCollection/fcselect?q={your question}&returnRSInput=true&generateHeader=true&fl=id)

`curl -X POST -u "{username}":"{password}" -F "answer_data=@retrieved_answers.csv" "https://gateway.watsonplatform.net/retrieve-and-rank/api/v1/rankers/{ranker_id}/rank"`

# Training Strategies

## Crowdsourcing

| Query   | Doc ID | Rank | Doc ID | Rank | Doc ID | Rank |
|---|--------|------|--------|------|--------|------|
| what similarity laws must be obeyed when constructing aeroelastic models of heated high speed aircraft.   | 184    | 3    | 29     | 3    | 31     | 3    |
| what are the structural and aeroelastic problems associated with flight of high speed aircraft.   | 12     | 4    | 15     | 3    | 184    | 3    |
| can a criterion be developed to show empirically the validity of flow solutions for chemically reacting gas mixtures based on the simplifying assumption of instantaneous local chemical equilibrium. | 236    | 2    | 166    | 2    | 488    | 0    |
| is it possible to relate the available pressure distributions for an ogive forebody at zero angle of attack to the lower surface pressures of an equivalent ogive forebody at angle of attack.        | 20     | 3    | 56     | 2    | 57     | 2    |

Define rank schedule (e.g. 4-0, perfect, very good, good, bad, irrelevant)

Challenges:

- SME bias
- Scalability

## Clickstream

learning to rank tutorial

Web Videos Shopping Images News More Search tools

About 1,020,000 results (0.49 seconds)

**Tutorials on Learning to Rank - Microsoft Research**  
[research.microsoft.com/en-us/um/.../tutorial.aspx](https://research.microsoft.com/en-us/um/.../tutorial.aspx) Microsoft Research  
 Tutorials on Learning to Rank. Tutorial talks. WWW 2007 tutorial on Learning to rank in vector spaces and social networks by Soumen Chakrabarti, Indian ...

**ACL-IJCNLP 2009 tutorial on learning to rank - Microsoft Research**  
[research.microsoft.com/en-us/um/.../acl-ijcnlp-2009-tutorial.p...](https://research.microsoft.com/en-us/um/.../acl-ijcnlp-2009-tutorial.p...) Microsoft Research  
 by H Li - 2009 - Cited by 4 - Related articles

**Ranking Methods in Machine Learning - SIAM**  
<https://www.siam.org/.../IT7A-LEARNING%20TO%20RANK%20TUTORIAL.p...> Society for Industrial and Applied Mathematics  
 Ranking Methods in Machine Learning. A Tutorial Introduction. Shivani Agarwal, Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of ...

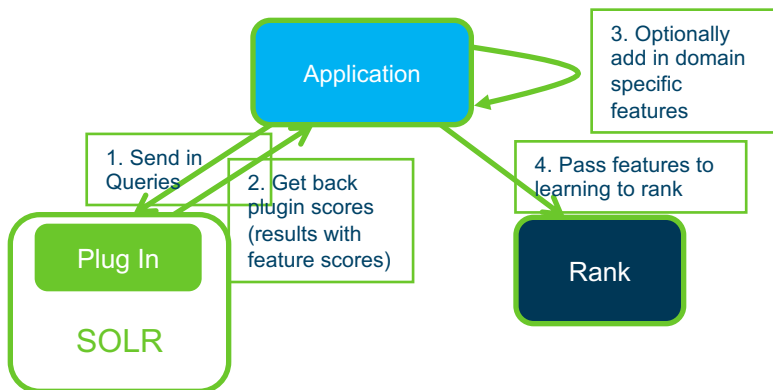
**Learning to Rank for Information Retrieval This Tutorial**  
[www2009.org/.../IT7A-LEARNING%20TO%20RANK%20TUTORIAL.p...](http://www2009.org/.../IT7A-LEARNING%20TO%20RANK%20TUTORIAL.p...)  
 Apr 12, 2009 - ... other conferences and journals might not be covered comprehensively. 4/20/2009. Tie-Yan Liu @ WWW 2009 Tutorial on Learning to Rank.

**Learning to Rank - Tutorial @ SEPLN 2012**  
[sepln2012.ijl.cs.cmu.edu/sepln2012\\_parrth\\_gupta\\_tutorial.pdf](http://sepln2012.ijl.cs.cmu.edu/sepln2012_parrth_gupta_tutorial.pdf)  
 Sep 5, 2012 - Introduction. Corpus. Eval Metrics. Approaches. Applications. Datasets. Letor API. Summary. References. Learning to Rank - Tutorial @ SEPLN ...

Active Learning Challenges:

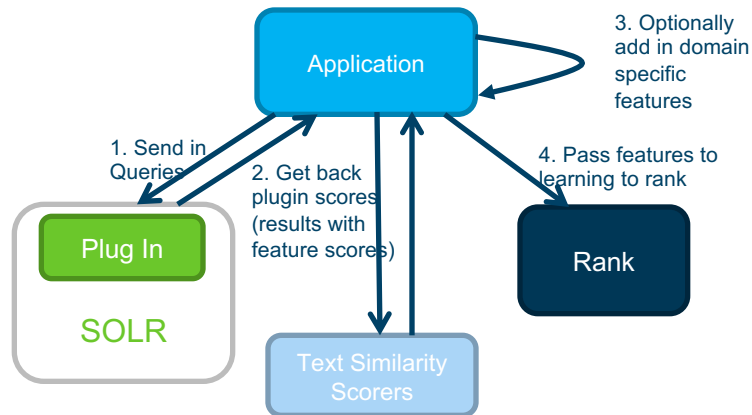
- Noisy
- Click decay based on rank
- Flip pairs
- Randomize results

# Extensible Decomposition



*Supports full feature set of Apache Solr (text search, dynamic clustering, faceted search, indexing...)*

*Whether these scorers are sufficient is dependent on application and use case. This is a reasonable place to start, but we may need additional text scorers.*



# Benefits

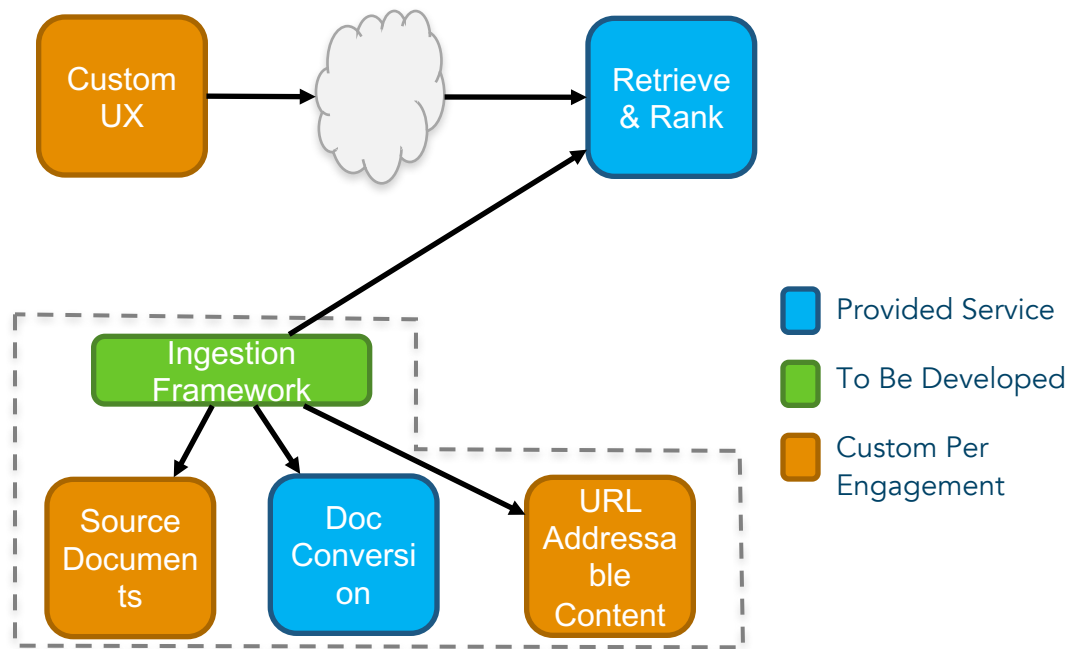
- Retrieve and Rank service improves standard information retrieval.
- Multiple ranked answers
  - Takes advantage of rich data in your documents to provide more relevant answers
- Large corpus/content
- Take advantage of external “metadata” about the documents to provide more relevant answers
- More control over training/ranking data
  - Irrelevant to relevant answers (non-negative integer)
- Custom features/scorers
- Benefits from community and Watson technology
  - New features developed both by the open source community
  - Advanced information retrieval techniques by the Watson algorithm teams.
- High availability
  - Each Solr cluster and ranker is highly available in the Bluemix environment.

---

# Architectural Patterns / Practices

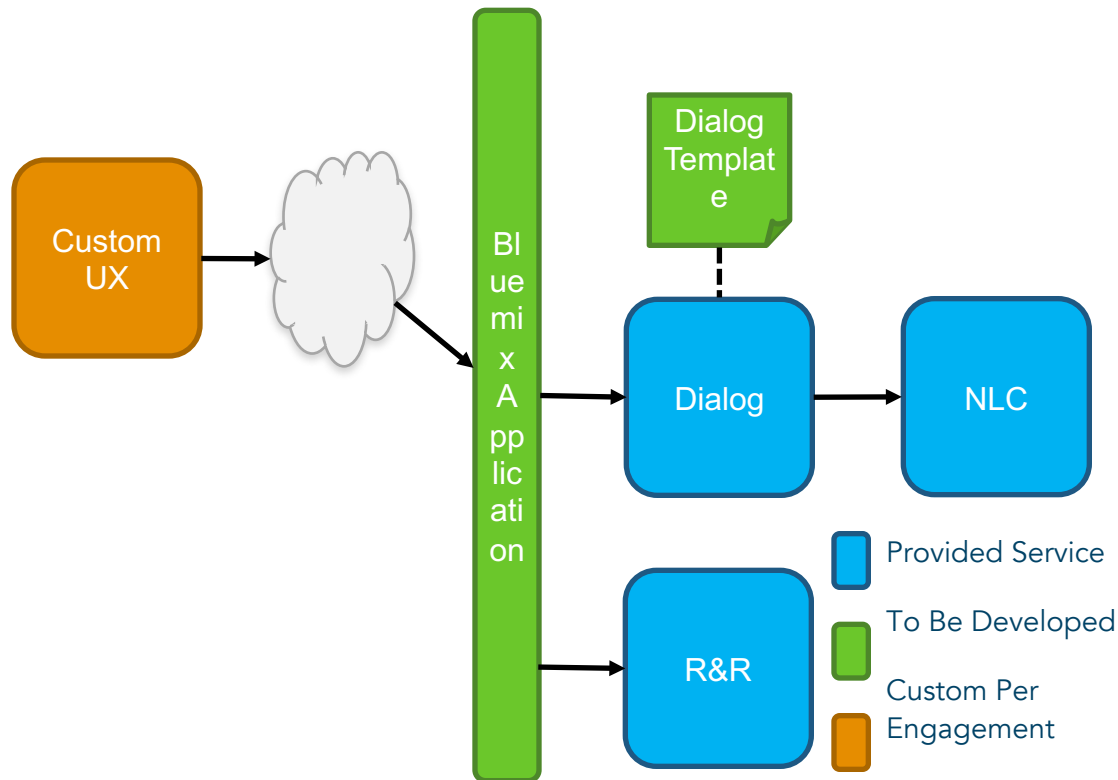
# Enhanced Retrieval Pattern

- R&R + Doc Conversion
- Minimum bar can be reached if:
  - Search index can be kept in sync with content
  - Training done via scripts supplied by the services
  - Content is available via a url
    - This may include a doc conversion step
  - Ingestion plan still to be worked out
  - No annotation



# Combined Conversation and Retrieval

- NLC, Dialog with R&R hand-off
  - Same as conversational, but with hand off to R&R
  - When confident in Conversational Pattern respond with confident answer
  - When not confident hand off to R&R



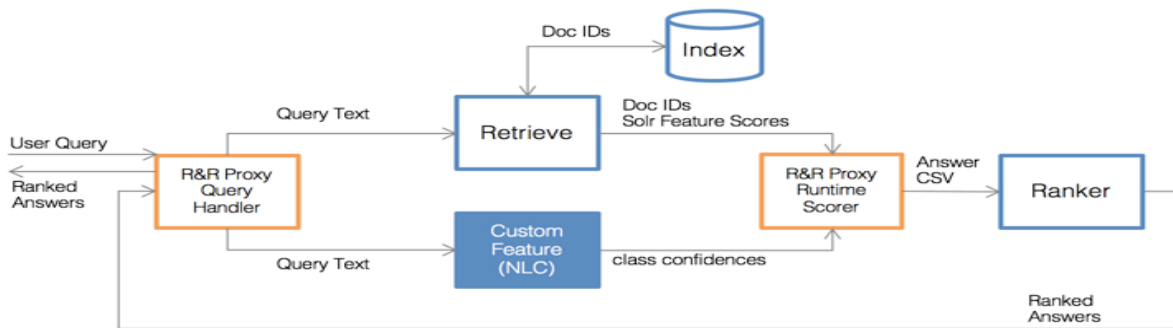


# Query Boosting w/runtime feature injection

For some R&R implementations, the native lexical features within documents are sufficient, but adding more domain knowledge through custom features often increases answer relevance and specificity. We can achieve this by extracting additional features not directly supported by Solr to inject additional cognitive training to our R&R system. This is an advanced feature that should likely be added after the other basic R&R implementation is in place.

There are many custom features we could create for R&R implementations but they fall into 1 of 3 categories: document, query, and query+document scorers. This blog post provides more details on injecting your own custom features:

<https://medium.com/machine-learning-with-ibm-watson/developing-with-ibm-watson-retrieve-and-rank-part-3-custom-features-826fe88a5c63#.9hybpgj5p>



# Tooling for Enhanced Information Retrieval

*Tools enabling developers to combine Retrieve and Rank and Document Conversion for a long-tail solution*

```
?? ??? ?????? ??? ????????
?? ?????????????? ????????
?????? ?????????????? ?????
?????? ?????????????? ?????
?? ?????? ?????????????????
?? ?????? ?????????????????
version 1.4.0

The command line administration tool for Enhanced Information
Retrieval.

The tool is used for the deployment of the Watson Document
Conversion service and the Watson Retrieve and Rank service.
Your collections can be created on Retrieve and Rank services
and crawler configurations can be created for the Data
Crawler to index document data into these collections.

Commands:
  kale login
  kale logout

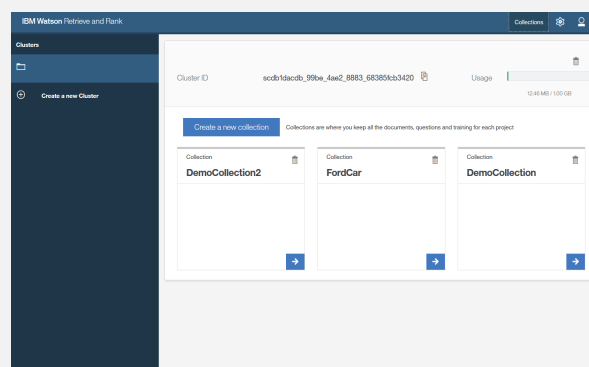
  kale create space <name>
  kale create document_conversion <name>
  kale create retrieve_and_rank <name>
  kale create cluster <name>
  kale create solr-configuration english|german|spanish
  kale create collection <name>
  kale create crawler-configuration
```

## Kale CLI

Setup, configure and manage services

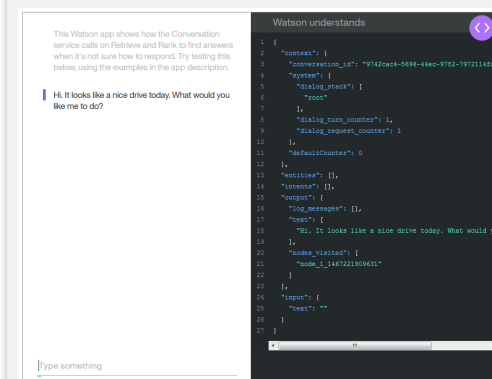
## Data Crawler CLI

Gather data from multiple sources to  
R&R



## R&R Web Interface

Speed up getting started ingesting documents  
and training with R&R



## Enhanced Conversation Sample App

Provide an example of connecting  
Conversation to R&R

# R&R Practices

- **Spellcheck is not enabled by default in R&R:** <https://brainsteam.co.uk/2015/11/17/spellchecking-in-retrieve-and-rank/>
  - You should consider enabling spellcheck in Solr/Retrieve&Rank to help users who have typos or misspellings in their queries/questions by suggesting corrections for them.
- **R&R query confidence**
  - A high confidence answer from the Rank side of side of R&R does not mean that the system is confident that the specific answer is correct. The confidence returned reflects how much better answer1 is to answer2 to answer3...etc, based on the trained model. Because that prior statement is possibly confusing, read this Watson Forum thread for more detail: <https://goo.gl/5Y2B8Y>.
  - Part of the reason for this is that it's almost impossible to get a normalized score from Solr . Please read this article for more details: <http://wiki.apache.org/lucene-java/ScoresAsPercentages>

---

**Thank You**