

IBM **Watson**

Retrieve and Rank Handbook

Links, Best Practices, Source Code, and Tools



Welcome to Retrieve and Rank!

This document will help you get started.

Page	Topic
	Introduction
3	Documentation, Community, and Code Libraries
4	Standard Workflow
	Best Practices
5	UI To Capture Ranked Ground Truth
6	Train user's to ask natural language questions
7	Obtain End User Questions ASAP
8	Enable Solr's Spell Checker
	Understand the Meaning of R&R Confidence
	Design Patterns
9	Combine Dialog + R&R + NLC
10	R&R for the "long tail" while NLC handles FAQs
11	"Fail-Over" to SME for Low Confidence Answers
12	Query Boosting w/Feature Injection

More Handbooks

Intro to Watson Development

<https://ibm.box.com/s/nav52vt6q2xwib5zqwupwif78mxtgems>


Natural Language Classifier (NLC) Handbook:

<https://ibm.box.com/s/rdlog2sue79178816s0rabkbi7ifu5vg>


Personality (PI) Handbook


<https://ibm.box.com/s/6h8dxsc3pq5idtgehjb6fwh7vbejsvtc>


Documentation and Community

 **R&R on the WDC:** <http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/retrieve-rank.html>
Read the R&R section of the Watson Developer Cloud to learn all about the service.


 **R&R on the Watson Developer Forum:** <https://developer.ibm.com/answers/topics/retrieve-and-rank/?smartspace=watson>
Read questions and get answers from other Watson NLC developers.


 **Technical Webinars:** <http://www.pages03.net/ibmwatson/building-with-watson-web-series>
Sign up for the upcoming webinars on how to build conversational apps using Watson


 **Blog: R&R Solr Configuration:** <https://goo.gl/ZspE1L>
Excellent post with a new dataset and ground truth for learning R&R

 **Blog: R&R Training and Evaluation:** <https://goo.gl/LfVnbK>
Part 2 of the prior blog post which now show training and testing your R&R solution

Code Libraries & SDKs

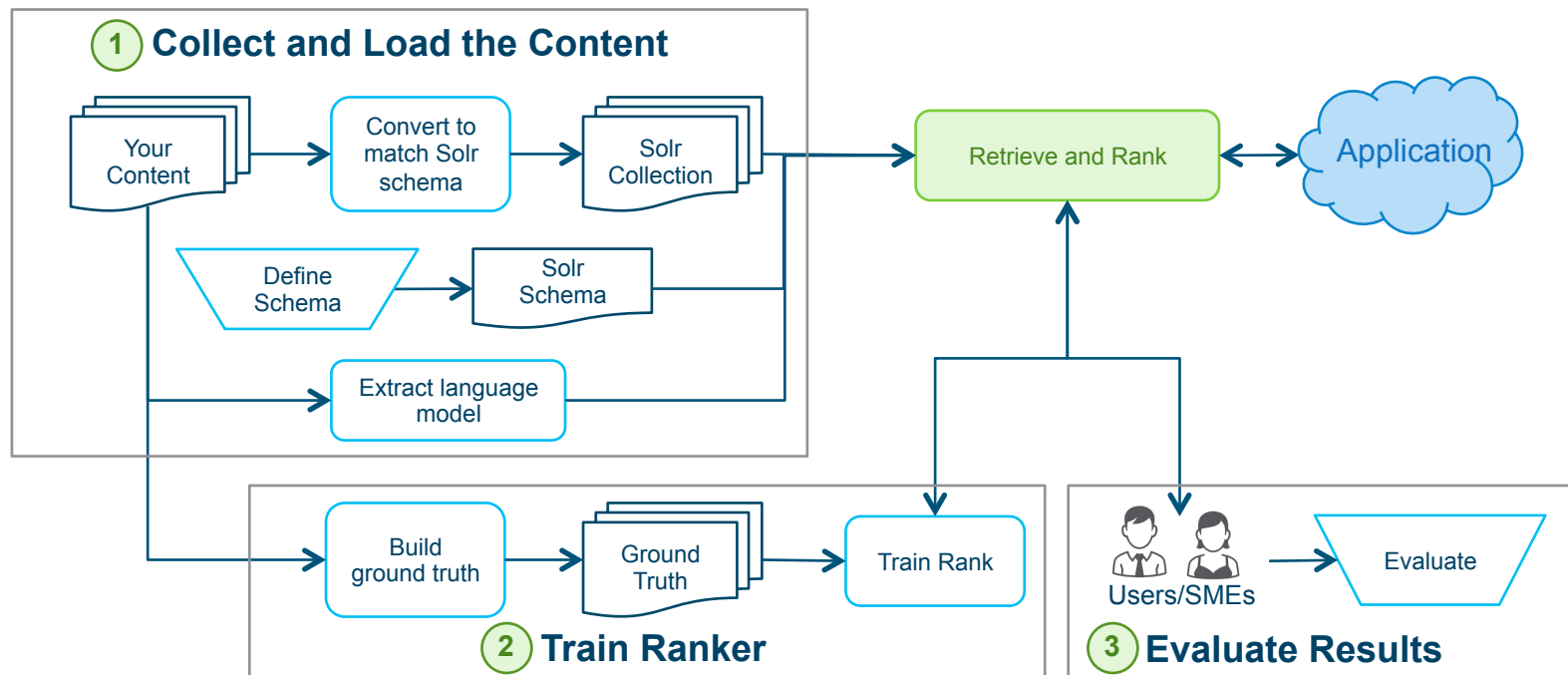
 **Watson NodeJS Library:** <https://goo.gl/W7RquQ>
Checkout this NodeJS library to kick start your Watson NodeJS development.

 **R&R Java Library:** <https://github.com/watson-developer-cloud/retrieve-and-rank-java>
Checkout this Java library to kick start your Watson Java development.

 **R Tools for R&R Ground Truth Analysis:** https://dreamtlearn.com/ryan/r_journey_to_watson/11
Get started analyzing your ground truth using R.

Standard R&R Workflow

The first question we hear is “How do I get started with R&R?” Here’s a rough overview showing the data collection, training, and evaluation steps. Each step is iterative so after evaluating results, you’ll typically collect and load missing content or improve your ground truth with additional questions.



Best Practice

UI To Capture Ranked Ground Truth

Why you need it: To train R&R, you will need to obtain a representative set of questions and a ranked list of great, good, and poor answers for them. This is the basic UX most popular for obtaining and capturing this ranked ground truth.

What it is: The example user interface below shows a simple way for how you could do so. This web page would display results from a partner's R&R instance and ideally push it to a database for later retrieval. As the button in the bottom-left indicates, you would need a way to rapidly export your captured ground truth in the correct format for import to R&R.

Store answer ranking to database
Each radio button selection should be automatically captured to the database (e.g. using AJAX calls).

Easy export of ground truth
Ensure you can rapidly export your ordered ground truth

Which line from Abe Lincoln's speech is most impressive? Search

Questions from watson@us.ibm.com

0 1 2 3 4 5

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war.

We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate -- we can not consecrate -- we can not hallow -- this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract.

The world will little note, nor long remember what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced.

It is rather for us to be here dedicated to the great task remaining before us -- that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion -- that we here highly ...

Export GT

1 2 3 4 5

Track each question submitted

Track the query progress of each user to monitor how user queries change during a session as they adapt to the answers returned

Capture each question's author

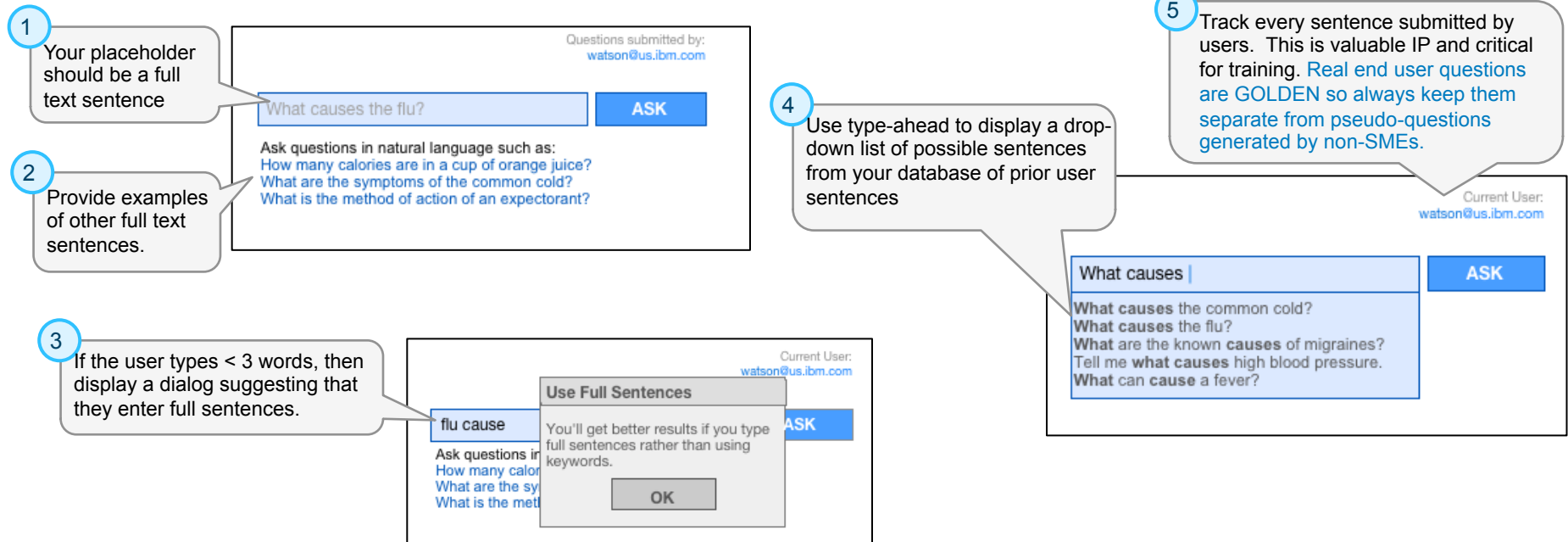
E.g. if I current user is different, click to change to the name. Associating questions to specific user lets you rapidly sort your ground truth by users inside/outside your company. **Real end user questions are GOLDEN** so always keep them separate from pseudo-questions generated by non-SMEs.

Best Practice

Train user's to ask natural language questions

Modify your user's default behaviors. No more keywords. Use full sentence.

If your UI has a text box, your users will likely default to entering keywords as they do with Google. You'll need to modify their behavior so they use full sentences. Here are a few recommended ways to achieve this.



Best Practice

Obtain End User Text/Questions ASAP**What's the first mistake most R&R implementations make?**

That's right. They spend too much time working with pseudo-questions generated internally to bootstrap R&R. Don't let yourself spend too much time before getting your application in-front of real end-users to validate your assumptions about how your users will ask questions.

Anecdotally, 99% of implementations waste 75% of their time generating ground truth that doesn't properly match end user needs. Validate soon and validate often!

“Remember that all models are wrong.
The practical question is how wrong do
they have to be to be useful?”



George Box, 1987

Don't wait for perfection. Present your users with “Good Enough”.

Pay close attention to George Box's quote and get in-front of your end users ASAP. Often a basic system is still useful enough to start asking meaningful questions. I.e. you need to know quickly understand your user's workflow, what questions/text they'll submit, and how it's worded. Normally a partial system will still be enough to validate and extend your initial assumptions.

Best Practice

Enable Solr's Spell Checker

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.

Best Practice

Understand the Meaning of R&R Confidence

What is meant by the “confidence” in R&R query responses?

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>.

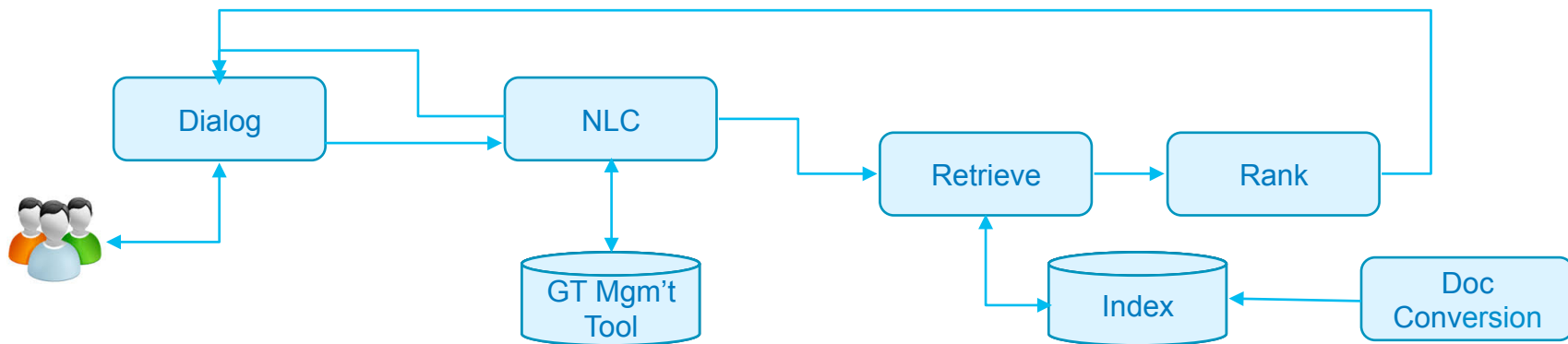
Part of the reason for this is that it's almost impossible to get a normalized score from Solr . Please read this article for more details:

<http://wiki.apache.org/lucene-java/ScoresAsPercentages>

Design Pattern

Combine Dialog + R&R + NLC

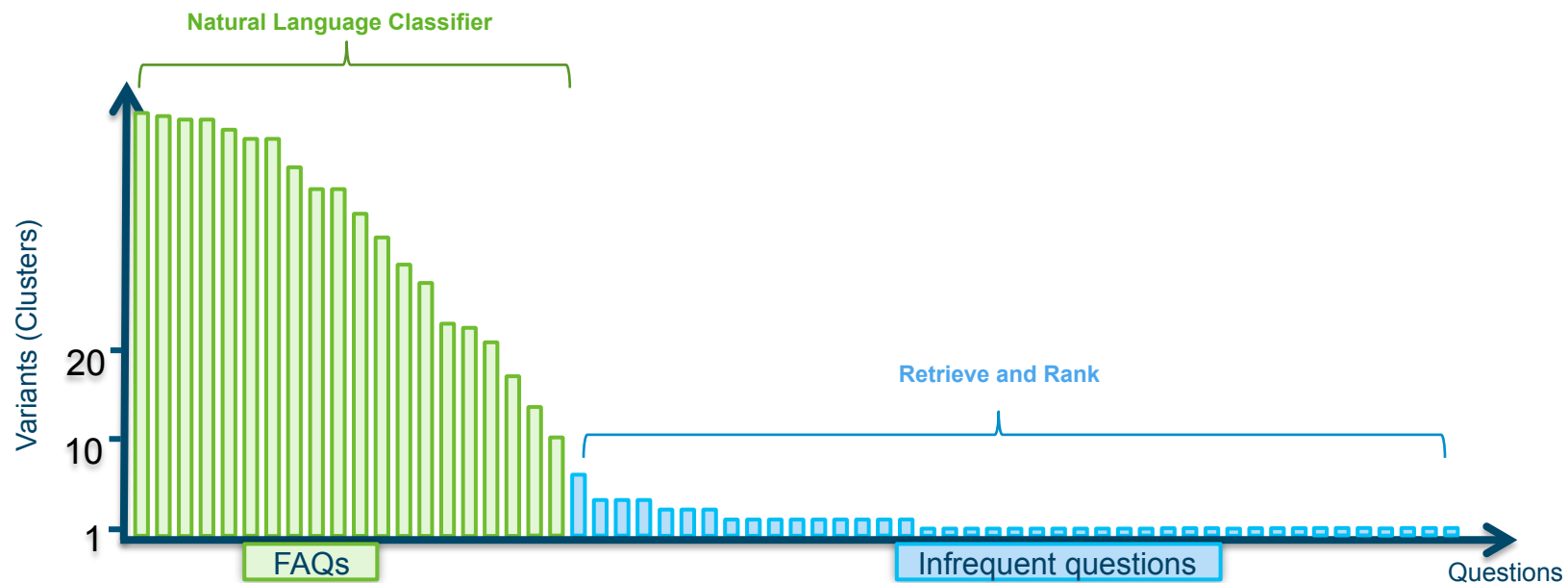
Combine Retrieve and Rank w/Dialog and NLC. Dialog provides the ability for a multi-turn experience where you asking clarifying questions to customers and track state across queries. NLC can be used to (1) detect specific domains of user interest so R&R can search only a subset of documents, (2) detect overlap between possible user intents so Dialog can request clarification by the user, or perhaps (3) NLC can be used to inject valuable run-time features into R&R for more targeted ranking of answers



Design Pattern

R&R for the “long tail” while NLC handles FAQs

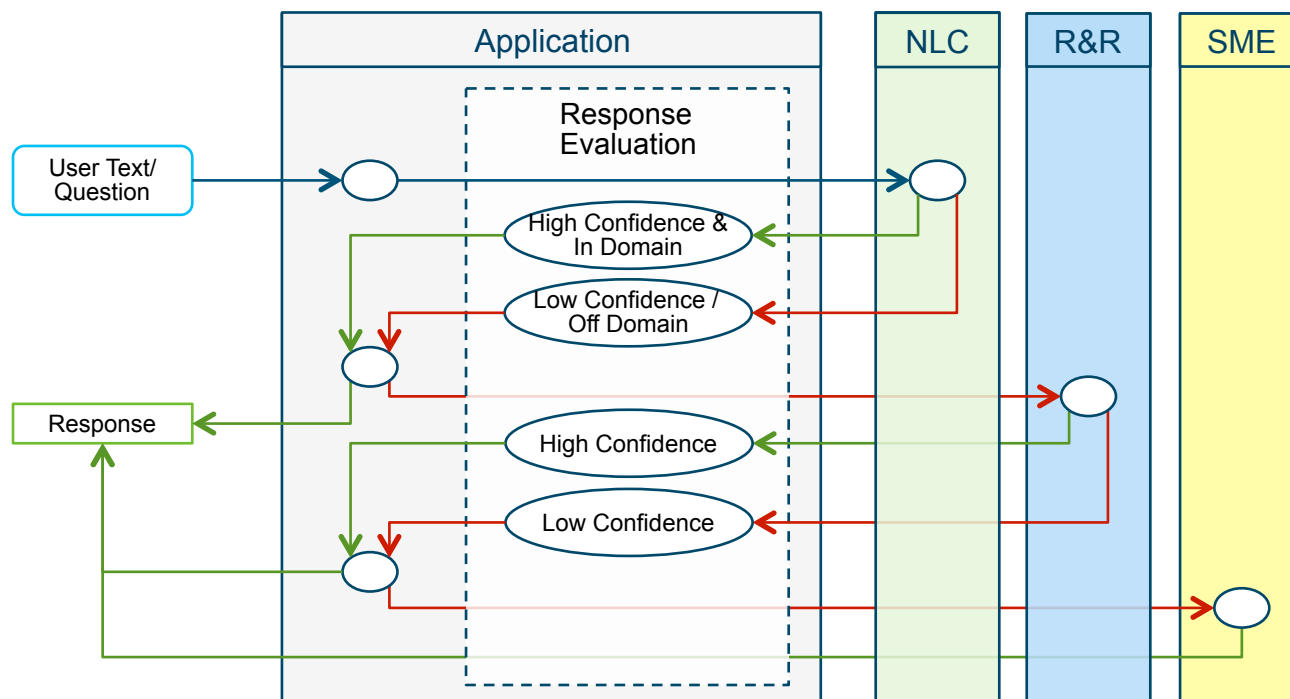
The **Natural Language Classifier** is well suited for Frequently Asked Questions (FAQs) where the effort to associate a single static answer to a question is rapidly rewarded. **Retrieve to Rank** is then used for infrequently asked questions or those for which multiple passages or frequently changing content must be searched.



Design Pattern

“Fail-Over” to SME for Low Confidence Answers

When applying the FAQ pattern using the NLC, it's common to fail-over first to Retrieve and Rank to determine if an answer can be found within the larger corpus of content. Or when even R&R fails to have the answer, user queries can be passed along to a subject matter Expert (SME) to process queries from high value customers.



Query boosting w/NLC-based 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:

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

Demo + Code: <https://github.com/watson-developer-cloud/professor-languo/>

More info on custom feature for Professor Languo demo: <https://github.com/watson-developer-cloud/professor-languo/blob/master/blog.pdf>

