IBM Watson
Natural Language Classifier

Links, Best Practices, & Design Patterns

Watson NLC: Tools & Community

NLC Homepage: https://www.ibm.com/watson/services/natural-language-classifier/
See the Developer Tools section of the NLC webpage

NLC on Stack Overflow: https://stackoverflow.com/questions/tagged/ibm-watson-cognitive
Read questions and get answers from other Watson developers

Code Libraries & SDKs: https://github.com/watson-developer-cloud
NodeJS | Python | Swift | Java | Unity | .Net | Salesforce

API Reference: https://www.ibm.com/watson/developercloud/natural-language-classifier/api
Learn how to effectively call the NLC API

Getting Started: https://console.bluemix.net/docs/services/natural-language-classifier/getting-started.html#natural-language-classifier
See the tutorial and the demo | Learn how to use your own data | See the release notes & interact with the developer community

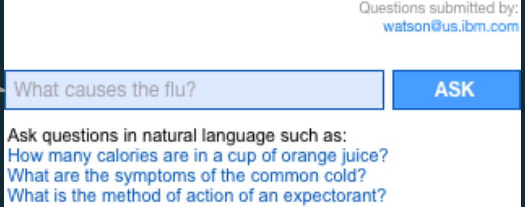
Best Practice

Train users to ask natural language questions

Modify your user’s default behaviors. No more keywords. Use full sentences.

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.

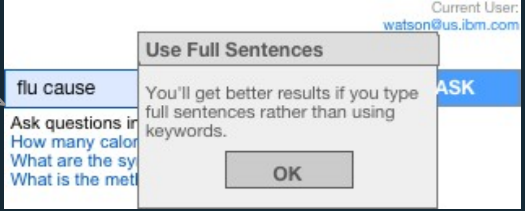
1 Your placeholder should be a full text sentence with examples


Questions submitted by:
watson@us.ibm.com

What causes the flu?

Ask questions in natural language such as:
How many calories are in a cup of orange juice?
What are the symptoms of the common cold?
What is the method of action of an expectorant?

2 If the user types less than 3 words, then display a dialog suggesting that they enter full sentences.


Current User:
watson@us.ibm.com

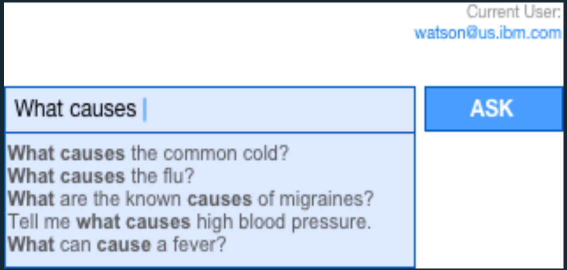
flu cause

Ask questions ir
How many calor
What are the sy
What is the metl

Use Full Sentences

You'll get better results if you type full sentences rather than using keywords.

4 Use type-ahead to display a drop- down list of possible sentences from your database of prior user sentences


Current User:
watson@us.ibm.com

What causes

What causes the common cold?
What causes the flu?
What are the known **causes** of migraines?
Tell me **what causes** high blood pressure.
What can **cause** a fever?

5 Track every sentence submitted by users. This is valuable IP and critical for training. **Real end user questions are GOLDEN so always keep them separate from pseudo-questions generated by non-SMEs.**

Best Practice

Best Naming Conventions for Intents

Take time to establish a consistent naming convention for your intents. First, determine a hierarchy that groups similar intents. We recommend a naming convention with the broader category at the left moving to subcategories on the right:

Text	**Intent**
Where is the fitness center?	fitness_center-location
Where is the business center	business_center-location
When does the fitness center open?	fitness_center-open_time
When does the fitness center close?	fitness_center-close_time
What are the business center hours?	business_center-hours

Separate categories by dashes and replaces spaces between words with underscores

Next look for subtypes that are common between intents. E.g. “location” and “hours” in the examples above. These subtypes could span multiple levels (as below), but we recommend not going beyond 4 levels in your intent hierarchy. Having good intent names will allow you to rapidly sort/search for all intents of a given type.

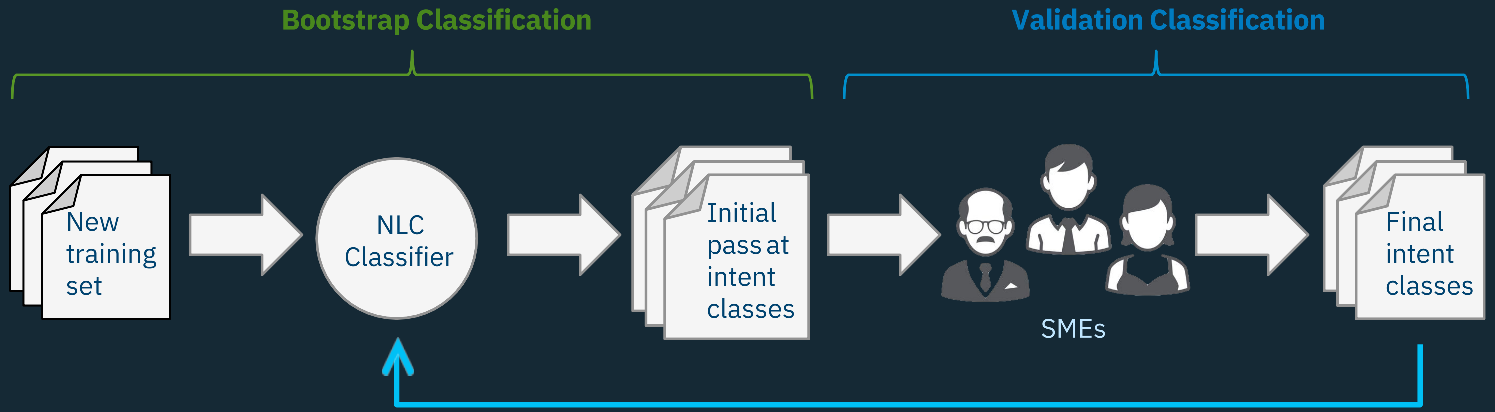
Can you send 2 pillows to my room?	housekeeping-linens-pillows
Just spilled coffee on my bed sheets. Help!	housekeeping-linens-sheets

Best Practice

Bootstrap your NLC Training

Are you tired of spending so much time training your NLC classifiers?

Wouldn’t if be great if prior training could speed-up the next round of training?



Try bootstrapping your next round of NLC training

Before you perform the next round of NLC training. Run your next set of training data against your current classifier and let it take a first pass at classification. Now go through this pre-classified training set and you only need to edit the classes that were mislabeled. Be extra careful not to simply accept the classes created by this bootstrapped classification. We don’t want the classifier to skew your training! Also pay attention to where your classifications are failing/succeeding as that can guide your future training efforts!

Best Practice

NLC Confidence Scores Sum to 100%

What confidence is returned by a classifier trained with only 1 intent class?

A single class NLC classifier would always return that intent class with confidence = 100% regardless of the text submitted. Let's see this with a simplified example where we train the NLC to recognize A and then submit X for classification.

NLC

If **A** Training → then given unrelated **X**, NLC will always return 100% confident the answer is **A.**

What about training with 4 intent classes then submitting input text that's absolutely not related to either?

So this time, let's train with W, X, Y, and Z.

NLC

If **A** and **B** Training → then given unrelated **X**, NLC will return 50% confident the answer is **A** and 50% it is **B.**

Similarly for more intent classes.

An NLC trained with 100 different intents classes would return confidence = 1% for each class when given text unrelated to the trained classes. This may sound like an odd usage of the idea of confidence, but it's incredibly useful. (See the next slide)

Trick Question: What if you trained the NLC on 10 classes *and NLC was 50% the answer was either of two of them*

It means the NLC is highly confident your text belongs equally to these classes. So you should consider this action: Review the semantic overlap of the classes to determine if they are really distinct

Design Pattern

Classify Complex Text with Multi-Intent Classification

How can we analyze text like: “Show me pink Audi convertibles”

We want to extract three intents from this text: color, model, and vehicle type. And we can do this by assigning three classes to our NLC training set. For example something like this:

Text	Class 1	Class 2	Class 3
Show me pink Audi convertibles	vehicle_color_pink	vehicle_model_audi	vehicle_type_convertible
Do you have blue BMW motorcycles?	vehicle_color_blue	vehicle_model_bmw	vehicle_type_motorcycle
Let me see pink BMW convertibles	vehicle_color_pink	vehicle_model_bmw	vehicle_type_convertible
…	…	…	…

Now use the fact that all NLC confidences always sum to 100%!

If the heading above doesn’t make sense, look at the Best Practices slide on “NLC Confidences Scores Sum to 100%.” That will help you understand this next conceptual example where going to train the classifer

NLC

If **A1**, **A2, B1, B2**, **C1** & **C2** Training then given **A1+B2+C1**, NLC will return 33% confident the answer is **A1,** 33% it’s **B2** and 33% it’s **C1**!

And the trick? Let A = vehicle colors, B = vehicle models, and C = vehicle types!

Yes. You’ve just trained your classifier to distinguish three intent classes with one call! You can now create a virtual vehicle sales agent that classifies user text by color, model, and vehicle type in a single NLC call. For our query above of “Show me pink Audi convertibles”, first separate the classes in your NLC response by each type (color, model, type) and normalize each intent category to 100%. This would allow you to determine vehicle_color=“”pink” at 100%, vehicle_model=“Audi” at 100%, and vehicle_type=“convertible” at 100%.

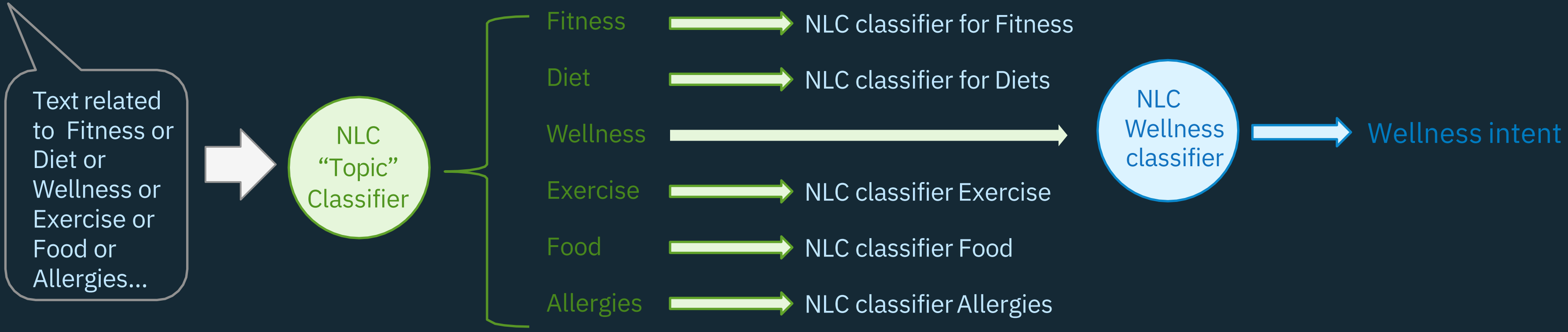
Design Pattern

Chaining Multiple NLC Classifiers

Do you have more classes than allowed by a single classifier?
Do your text classes have higher level categories that are easily separated?
Is training a single classifier becoming increasingly difficult to manage?

Then perhaps a multi-classifier solution is a solution for you:

The initial classifier would perform a high-level separation of your text so that classifiers at the next level can separate your classes with higher confidence.



Text related to Fitness or Diet or Wellness or Exercise or Food or Allergies…

NLC “Topic” Classifier

Fitness → NLC classifier for Fitness

Diet → NLC classifier for Diets

Wellness → NLC Wellness classifier → Wellness intent

Exercise → NLC classifier Exercise

Food → NLC classifier Food

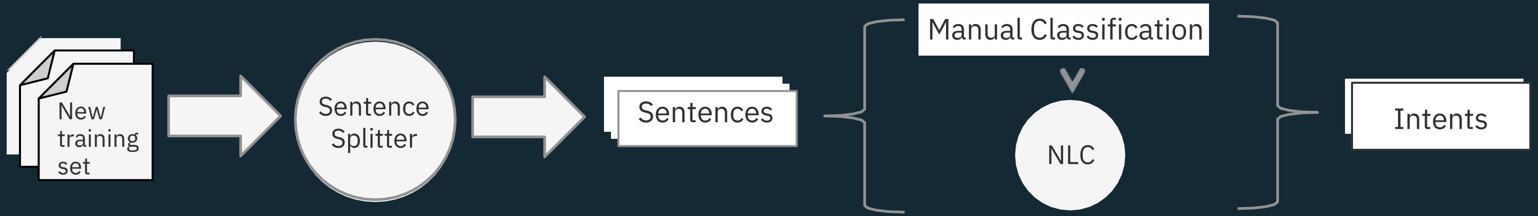
Allergies → NLC classifier Allergies

Design Pattern

Use NLC On Large Datasets Through Decomposition

At the moment, NLC works with sentences <1,024 characters so it won't directly work with long pieces of text. Long text can be broken into sentences - classifying each sentence and then aggregating the result into a summary of the message. There are a number tools to accomplish this such as the Watson Natural Language Understanding Service.

The training phase requires manual annotation of a set of test documents (start with a couple of hundred - it should take a couple of hours to plow through them and assign a class for each sentence).



The production flow will use the trained classifier to automatically do the sentence level classification and then spit out a summary - you can use business rules to trigger actions at this point.
For an example use case where we detect the severity of complaint emails and rank them from minor to severe:

If the email contains 10% or higher "severe" sentences then prioritise triage.
If the email contains 90% "minor" sentences then leave at bottom of queue
If the email contains 90% "severe" sentences then email the CEO and assemble the press team

Or train a meta-classifier (Decision tree or SVM) to make a classification based on the results from the sentence level annotations.