Watson Assistant Continuous Improvement Best Practices

Authors:
Adam Benvie - Offering Management, Watson Assistant
Eric Wayne - STSM and Development Manager, IBM Watson
Matthew Arnold – IBM Research staff member

Introduction

This document details the best practices for the on-going management of your Watson Assistant based virtual assistant after it has been deployed to production. It defines the steps and concepts integral for you to effectively and continuously improve your virtual assistant. The recommendations you'll find in this document are based on successful client experiences from experts in the virtual assistant space.

Business KPIs are King

No matter how engaging or accurate your virtual assistant is, it should be making a positive contribution on your business. There are many potential ways you can expect your Watson Assistant to achieve results, and it's important to know which ones are aligned to the goals of your individual business.

Your specific business objective should be identified, along with the metrics you'll use to measure its success. For example, if your goal is to reduce customer service costs, it should cost less to maintain your assistant than it does to staff an equivalent customer support team.

The tactics discussed in this document outline how to measure and improve how well your virtual assistant serves your customers. However, keep in mind that this document does not recommend business KPIs, as these can only be defined by you and your business.

Continuous Improvement Process

In order to operate a virtual assistant which continues to meet all your objectives, it is imperative that you have a formal process in place which is thoroughly executed.

An effective improve process will meet the following criteria:

(1) Provide a reliable understanding of overall performance of the virtual assistant
(2) Clearly show where you should be prioritizing your improvement efforts
(3) Allow you to make the necessary improvements as efficiently as possible

The six improvement phases here in Figure 1 make up a process that meets these criteria.

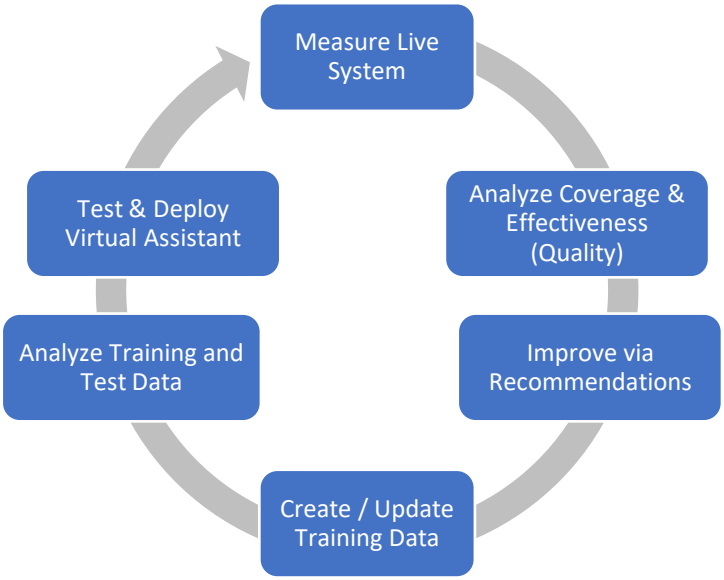

Figure 1: Continuous Improvement Process

Resources for Continuous Improvement

The Watson Assistant service provides a variety of features to help you continuously get the most out of your virtual assistant. These features include improvement recommendations from Watson, as well as analytics to enhance the conversations your assistant is having with customers. Additionally, IBM has published Jupyter notebooks that help you measure your assistant's performance and analyze problem areas to improve. These resources will be referenced throughout this document to help you understand how to use them for continuous improvement. Please see Appendix B for a summary view of how the four Jupyter notebooks apply to the phases of the improvement process.

Create Training Data for Initial Deployment

While this document focuses most on improvement after production deployment, this section introduces concepts that you can leverage from the start of your project to best position yourself for ongoing improvement. The very first step of the continuous improvement process is to create training for your assistant before you deploy it to your users in production. You'll only perform this initial training step once. Afterwards, you'll use what you learn from your user interactions to improve the training. As you begin, your awareness of these improvement best practices will help you get off to a good start.

First, consider the set of intents that you want your assistant to support. These may be questions that are most frequently asked or tasks that are the highest value to your business. Don't try to "boil the ocean" for your first version.

If you are adding your assistant to a use case where you already have a live chat system, you can use the chat transcript logs from existing systems to guide and prioritize your choice of which intents to cover. Through the Intent Recommendations feature, Watson Assistant analyzes your chat transcript log data to find the most common and frequently expressed customer needs related to your business. Watson recommends intents and intent user examples that you can use to train your assistant so it will recognize similar requests in the future. This step not only saves you time, but also helps you start with greater confidence that your assistant is covering common areas.

Consider the primary ways users will ask questions or request tasks, and add these as examples to each intent definition. User example recommendations will help you find different ways the intent is expressed in your logs. You might also conduct a survey of your user audience to find additional forms of user examples per intent. In the first version of your chatbot, target 10-20 user examples per intent. If you know your intent will cover an especially broad scope, you may need to add more than 20 examples. The Intent conflict resolution feature will help you identify and resolve possible confusion between intents by finding user examples in more than one intent that overlap.

For each intent, flesh out the dialog steps required to gather information, perform transactions and provide responses to your users. Also identify which dialog steps and conditions would mark successful completion of the requested task. For example, if the task is to open a new account, the confirmation of the creation of the account after gathering information from the user would indicate successful task

completion. Identifying task completion will help you measure the success of your chatbot.

Use entities to help you craft more useful, targeted responses. For example, you might define a task to purchase a product. When a user makes a request that triggers that task, your assistant's response should reflect an understanding of what the customer wants to buy. You can add a product entity, and then use it to extract information from the user input about the product that the customer is interested in. The Entity synonym recommender may help you expand your entity definitions with more ways that a user can refer to the entity.

Create "placeholder" dialog nodes for topics that you know users will ask about, but you aren't ready to support. This allows you to provide a prescribed response, and as you later make improvements, you'll be able to distinguish these topics from others that you inadvertently did not train the chatbot to handle.

The first version of your assistant does not need to be perfect. You can rely on the Disambiguation feature and fall back options of Watson Assistant as means of flexibility when your users express an utterance that you haven't yet accounted for in your training. The disambiguation feature allows your assistant to ask customers to clarify their meaning when the assistant isn't sure what a user wants to do next. When a given input matches multiple dialog nodes, the assistant shares a list of the top node options with the user and asks the user to pick the right one. If you are using the Web Chat integration your assistant's experience will also include the ability for a user to click to see more possible responses if the main response does not meet their need.

You can also choose to include "fall back" options in your dialog design. Consider how you want your chatbot to respond when it has not yet been trained to answer a user's question. You can use the "anything else" dialog node to give a specialized message such as "I'm sorry, I'm not yet trained on that." Or, you might configure a Search skill to crawl and then search a knowledge base for an answer when your assistant is not confident enough to give a response.

Another fall back option is to transfer the user to a live human agent. If the user's input matches an intent or other dialog node conditions, you can ask them if they would like to speak to a human agent, and then transfer them using Watson Assistant's Service desk platform integrations.

Before you deploy, spend time testing with a set of utterances that are not part of the training. Testing can build your confidence that your intent, entity and dialog node definitions are well formed. Later as you change your training, tests help you detect problems that would break the assistant functionality. If your test results do not meet your expectations, you can use the Dialog Skill Analysis tool to help identify problems with your training or test data before you deploy.

Go live with your first version. The remainder of this document will help you chart your progress and guide you to where you can best spend your limited time to help your chatbot improve.

Measure Live System

After you've deployed your assistant to production, you'll want to quickly begin to measure its interactions with users. The goal at this stage is to understand the areas in which your virtual assistant is doing well, versus where it isn't.

Effectiveness and coverage are the two measures that provide the most reliable understanding of the overall performance of your assistant, and the combination is very powerful for diagnostics. If your coverage and effectiveness metrics are high, it means that your assistant is responding to most inquiries and responding well. If either is low, the metrics provide you with the information you need to start improving your virtual assistant. You can implement these measures by using the Watson Assistant Measure Notebook.

Measure Effectiveness (Quality)

The first measure which should be the focus of your improvement effort is *effectiveness.* Effectiveness measures how well your assistant is handling the conversations or messages it is responding to. This includes any automated quality measures that help identify problematic conversations, such as:

Conversation Containment

Conversation containment measures the portion of conversations not handed off or escalated to a human agent for quality reasons. Let's assume one of your assistant's objectives is to handle conversations otherwise handled by humans. If you design your assistant to address a customer's need and the assistant is unable to fulfill the task and hands the conversation to a human, then this conversation is not contained and can be deemed an ineffective conversation. You can measure containment using the "Connect to Human Agent" response type in Watson Assistant's Dialog, or by training an intent with associated dialog logic to hand off to a human agent when customers ask for one. Whenever a conversation hits a "Connect to Human Agent" response type or intent, the conversation has not been contained.

Task Completion

Task completion measures whether or not your assistant completed the task requested by the customer during a conversation. You may also think of a task as the journey your users take through a sequence of steps to obtain what they need. Tasks can be defined as either transactional or informational. An informational task simply refers to providing requested information. You can measure informational tasks by prompting customers to tell you whether or not the provided information was helpful. Conversations where a customer receives information but marks it as not helpful can be deemed ineffective.

A transactional task refers to helping a customer through a process during a multi-turn conversation, such as booking an appointment. You can measure transactional tasks by labeling dialog nodes that mark the beginning and completion of a task. If an intent trained to initiate a task was hit, but the completion node was not, then the conversation can be deemed ineffective.

NPS

NPS is a widely adopted measure of customer satisfaction that has been shown to correlate with revenue. You can measure NPS at the end of a sample (or all conversations) by prompting users to rate their experience on a 1-10 scale. Ineffective conversations defined by NPS would be those which received a rating of below seven.

Measure Coverage

The second important measure is *coverage*, which measures the portion of total messages your assistant recognizes and attempts to respond to. In general, this measures those inquiries which your assistant provided a real response, versus those it did not (e.g. it responded "Sorry, I'm not trained on that"). Coverage is a measure of how often your virtual assistant thinks it has a response, whereas effectiveness is a measure of how effective those responses are. The distinction is helpful because the process of improving your virtual assistant is different for questions your assistant is answering, versus those it isn't.

Coverage is measured as a percentage of total messages for which the assistant returns a real response. You can measure coverage by leveraging intent confidence thresholds and anything_else dialog nodes. If a message does not hit the default intent confidence threshold (0.20), and is marked as Irrelevant or does not hit your custom set confidence threshold, the message is not covered. Additionally, if it hits an "anything else" node, it is also not covered.

Deciding what your virtual assistant should or should not cover is a critical decision you and your team needs to make. For example, if you're creating an assistant to help customers obtain their credit score, you should not spend time training your assistant on how to answer questions related to credit card interest rates. However, if you see that customers are frequently asking about certain topics your assistant is not designed to cover, you should train it to respond to these topics in a way that communicates its coverage scope to the customer.

For example, you could train your assistant to respond generally to certain inquiries with "I understand you are requesting interest rate information but unfortunately I cannot handle questions on this topic. For interest rate information please follow the link below." Messages triggering these responses should be measured as covered.

Measure Effectiveness (Quality) & Coverage

Analysis of effectiveness and coverage should complement each other. If your assistant is answering questions incorrectly, you can improve it by focusing on improving effectiveness. If your assistant is not answering enough questions, you improve this by focusing on expanding coverage. For any given effectiveness metric (such as containment), you can view coverage separately for both the escalated and not-escalated conversations. This will allow you to get more insight into where you should focus your attention.

Consider the Effectiveness and Coverage metrics as reported in Figure 2 from the Watson Assistant Measure Notebook, where containment is our measure of effectiveness. In this scenario, 73% of conversations have been escalated to a human agent. There are many conversations that are not being contained, which means you should focus on these conversations. Furthermore, within these escalated conversations, the assistant has responded to a significant number of questions (84%), which suggests the effectiveness of the responses is likely the problem, not a lack of defined responses (i.e. coverage). With this understanding, you should perform a deeper analysis on the covered messages within the escalated conversations.

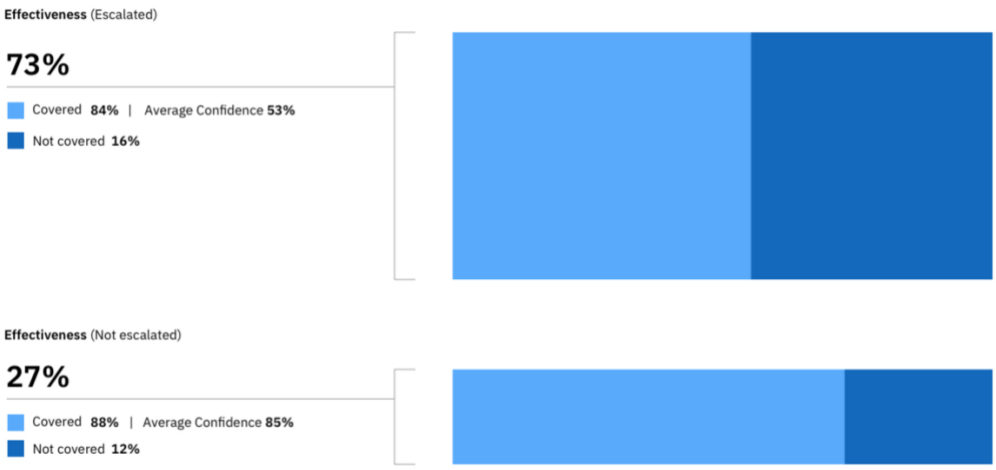

Figure 2: Effectiveness & Coverage

The Watson Assistant Measure Notebook helps you narrow your immediate focus of improvement by looking at effectiveness and coverage measurements. You can choose to improve both the effectiveness and coverage dimensions, but they're most often improved as separate processes. The following sections outline steps toward analyzing and improving effectiveness (quality). From there, we'll outline steps for analyzing and improving coverage.

Analyze – Effectiveness (Quality)

If you've decided to analyze your assistant's effectiveness, you'll want to perform a deeper analysis of the group of ineffective conversations you identified from the Measure Live System phase. Following the example from Figure 2, this analysis would be on the covered messages from escalated conversations. Figure 3 illustrates that after you've decided to focus on improving low quality conversations, you then choose between two primary focus areas:

1) Problems with detecting the right intent or entity
2) Problems with the flow of the dialog after the intent or entity was detected

You may have evidence from customer feedback to suggest which of these two areas requires the most attention. Alternatively, you may need to spend time in each area to uncover root causes of problems. The following sections describe each of these paths in greater detail. Notebooks provided by Watson Assistant help you identify root causes of problems and determine the set of actions you need to take for improvement.

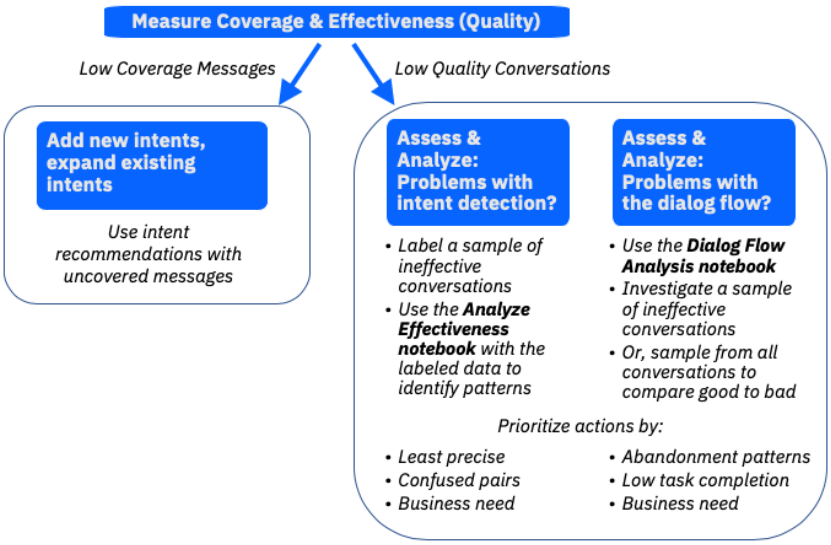

Figure 3: Focus Area Decisions

Analyze the Quality of Intent and Entity Detection

This analysis is performed with three steps:

(1) Sample a random set of utterance-response pairs from ineffective conversations you're analyzing
(2) Manually assess the sampled logs
(3) Analyze the results with a notebook to create list of improvement actions

Sample

Question-response pairs should be randomly sampled from the group of ineffective conversations you chose to investigate in the Measure Live System phase. In the above example, this would be a random sample of the covered messages from the not-contained conversations. The Watson Assistant Measure Notebook provides an easy way to export a spreadsheet of sampled question response pairs with columns for recording the assessment.

Perform a Manual Assessment

For each sampled question-response pair:

(1) Manually determine whether the response was:

Correct or incorrect

Was the response returned by your assistant the response you expected it to give? The percentage of correct responses tells you the *precision* of your assistant.

Helpful or unhelpful

If the answer was correct, also mark whether it was *helpful.* The goal is to identify responses that may be considered technically correct, but the wording of the response is not satisfying to the user. It could be too long, too general, or just worded awkwardly, thus resulting in an overall ineffective response. The exact definition of helpfulness is subjective and can be defined based on your business goals.

The distinction between precision and helpfulness here is important because it helps determine where your assistant needs improvement.

(2) Specify the correct system response for each utterance your assistant responded to incorrectly.

Identify the root cause by marking whether it was due to an incorrect or absent intent, entity, or dialog response.
Specify what the correct intent, entity, and dialog responses should have been.

a. If the required intent does not exist, mark it as "new intent needed"
b. If the required entity does not exist, mark it as "new entity needed"
c. If the required dialog logic does not exist, mark it as "new dialog logic needed"

Analyze the Results

Feed the manual assessment results into the Analyze Effectiveness Notebook and view the analysis. The notebook will help you understand the relative performance of each intent and entity as well as the confusion between your intents. This information helps you prioritize your improvement efforts.

Summary Metrics

The Analyze Effectiveness Notebook shows you the following summary metrics based on your assessment:

- The overall precision and helpfulness of your assistant's responses
- The number of utterances assessed
- The number of correct responses
- The number of incorrect responses

Note that the precision is reported together with a confidence interval based on your sample size (see Appendix A). You should always use these intervals when comparing assessment results, with the understanding that changes are not statistically significant if the confidence intervals overlap. You can reduce confidence intervals by increasing the size of your sample.

Intents

Intent analysis is performed by looking for the dominant patterns of intent errors. One effective way to perform this analysis is to focus on four main categories of errors:

(1) Worst overall performing intents. These are intents that are most involved in a wrong answer, whether due to precision or recall.

(2) Worst recall intents. This identifies intents that are being missed (for example, failing to classify utterances to itself) most often. This intent should have been given out, but other intent(s) are matching instead. These intents likely need more training examples.

(3) Worst precision intents. This identifies intents that are frequently matching even though they should not be, thus hiding the correct intent(s). These intents likely have training examples that clash with the training of other intents.

(4) Most confused intent pairs. These are pairs of intents that are often confused with each other. These intents likely have training examples that overlap. Examples of confused intents pairs are seen here in Figure 3, courtesy of IBM's Analyze Effectiveness Notebook.

Total Wrong	Intent 1	Intent 2	Incorrectly in Intent 1	Incorrectly in Intent 2
668	capabilities	turn_on	18	650
18	locate_amenity	turn_off	11	7
16	greetings	locate_amenity	9	7
13	capabilities	locate_amenity	8	5
12	locate_amenity	turn_up	8	4
9	locate_amenity	weather	3	6
7	locate_amenity	out_of_scope	2	5
5	locate_amenity	positive_reaction	2	3
5	greetings	weather	2	3

Figure 4: Confused Intent Pairs

Entities

The Analyze Effectiveness Notebook will highlight any entities and entity values that are particularly problematic so they can be targeted for further investigation and improvement.

Analyze the Quality of Dialog Flows

Now you are ready to investigate problems with the flow of the dialog after the intent or entity was detected. Flow-oriented visualizations can help you analyze task completion and user journeys in the context of your dialog steps. For example, you can see patterns such as higher frequency of *abandonment* where the user stops engaging with the assistant at specific steps in the flow. You can use the Dialog Flow Analysis Notebook to drill down and investigate **where** and **why** the assistant is losing engagement along the dialog flow.

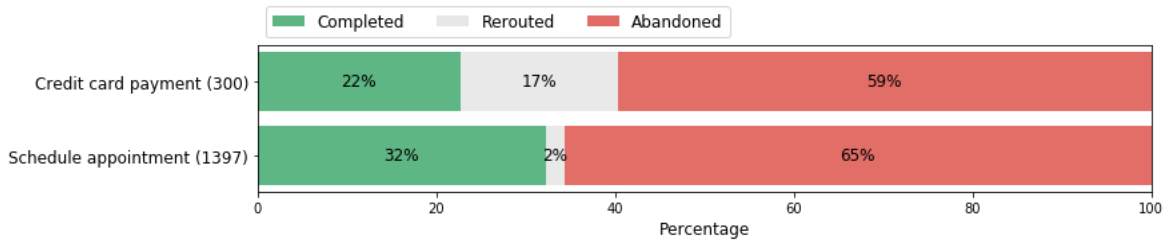

Figure 5 – Measuring task completion rates.
*The **Schedule Appointment** task has high volume and low completion rate*

The dialog flow visualization is an interactive tool for investigating user journeys, visits and abandonments within the steps of the dialog logic. The sequences of steps from every conversation are aggregated to provide

a view into users' journeys at the level of all conversations or a specific user segment or period of interest. You can use the visualization to understand entire journeys or focus your exploration on particular sub-journeys or key steps of interest, through the concept of *milestones*.

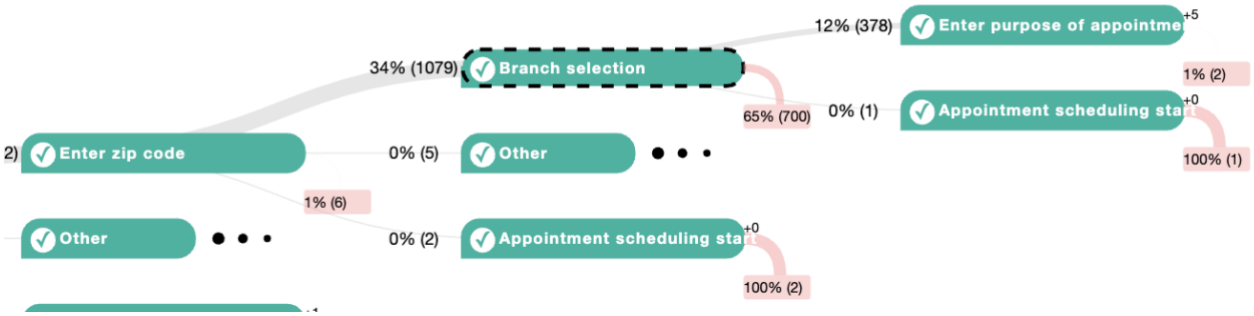

Figure 6 – An example of a milestone-based dialog flow visualization.
*The visualization can be used to identify common patterns of failed conversations, for the **Branch selection** step which has a relatively high drop off rate in the middle of the flow.*

When you click on a node, you can see the total number of conversations that progressed along a particular sequence of steps, and the number and ratio of conversations that ended at that step. Detecting frequent conversation sequence patterns that lead to user drop off (abandonment) at an unexpected location in the dialog can help you find **where** users lose engagement. You can select a group of conversations that fit that pattern to further analyze and improve.

Next you can drill down into user utterances and conversation steps in order to detect potential common issues that might have led to user drop off. Some possible reasons could be:

- The assistant didn't anticipate or understand the user utterance at that point of the conversation
- The options presented to the user were not relevant
- The conversation didn't progress in the right direction
- The assistant's response wasn't accurate or relevant

The Dialog Flow Analysis Notebook helps you investigate those conversations using two visual components. The **transcript** analyzer lets you investigate the full sequence of conversation steps in the selected group of conversations. You can view the user utterances as well as observe the assistant response and the dialog nodes that were triggered to produce those responses. This allows you to understand the conversation in context and pinpoint the exact source of issue.

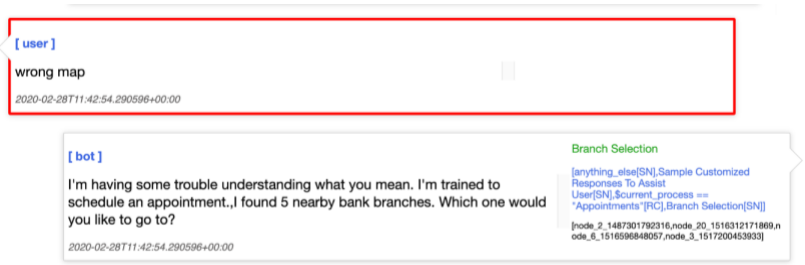

Figure 7 - Transcript visualization. The visualization can be used to understand user utterances and assistant respones in context, and to pinpoint issues that lead to abandonment

The **keywords and phrases** analysis helps you identify common terms that are frequent and representative in the group of selected conversations. You can also compare terms in failed versus successful conversations in order to find terms that correlate with failed conversations. Investigating terms in failed conversations can help you to quickly assess if you should focus on your intents and entities or your dialog flow design.

Figure 8 - Keywords and phrases analysis.
The visualization shows which terms are correlated with failed conversations

In addition to detecting user abandonments, you can also use the dialog flow visualization to measure the distribution of flows in your dialog logic or compare flow ratios between subsequent releases.

Improve – Effectiveness

After completing your assessment and analysis, the next action to improve your assistant's effectiveness is to update your workspace based on the results of the assessment you performed in the Analyze phase. For any intent, entity, or dialog response that was incorrect, the corrected version should be added to training when appropriate.

For example, if the intent was incorrect, you can add the user message to the training examples for that intent. If the entity was incorrect, the entity training can be updated to handle it properly in the future. If the dialog logic had issues, it should be corrected.

If the intent, entity, or dialog node that is needed does not exist, then you should consider adding it. This process is described in more detail in "Improve – Coverage" section of this document.

After those updates have been performed, you have the option to continue focusing your improvement efforts on any trouble spots identified during the Analyze phase.

Intents

Problematic intents can be improved with the following series of techniques:

(1) Fix existing training that is causing confusion between intents
(2) Add training to confused intents to clarify their boundaries

(3) Add training to imprecise intents
(4) Combine the confused intents into a single intent and distinguish using entities

Fix Existing Training

To eliminate confusion between intents, look for mistakes in training data that may be causing the confusion. The IBM Watson Assistant Conflict Resolution feature helps you find training examples that may have been added to the wrong intent and are confusing your classifier. Use this to help you identify user examples to resolve. If you see a significant number of conflicts between intents and you're unable to resolve them, your intents may be too nuanced and you should consider combining them into one intent, as described further below in the "Combining Intents" section.

Adding Training to Clarify Boundaries

Once conflicts between confused intents have been fixed, the next thing you should do is add training designed to help clarify the boundary between intents. In other words, distinguish the two intents from each other. This is done by adding utterances that the classifier thinks could be either of the confused intents. More precisely, these are utterances where the two confused intents are ranked first and second, and the confidences have a small delta (< 0.10). As you add these utterances to the correct intent, the boundary between intents will become clear.

Adding Training to Imprecise or Poor Recall Intents

During your analysis, you may find intents that are particularly inaccurate and confused with a range of other intents, rather than confused as a single pair. To improve this scenario, first resolve all conflicts within the problematic intent, as described above. Second, add training examples to help remove the confusion. If the problematic intent has *low-recall* (it is not being selected as often as it should), then training should be added directly to the intent to help it be selected more often. If the problematic intent has *low-precision* or *recall* (it is given out when it shouldn't be), then training should be added to the intents that are being overshadowed by the problematic intent. The User example recommendations feature helps you add training to a specific intent.

To maximize the impact of adding training to an intent, you should avoid adding utterances that have already been matched to the intent you're training with a high confidence (i.e. above 0.80). The intent is clearly not struggling with these examples. It's better to add examples that were matched at a low confidence, so the intents will correctly classify similar utterances in the future when it previously did not.

Combining Intents and Using Entities

If you have intents that you're struggling to differentiate using the approaches above, you should evaluate whether the intents are too narrow or nuanced. Consider merging them into one intent, using entities to differentiate the appropriate dialog response. For example, if you have intents #open_credit_card and #open_debit_card, it is best to combine these as a single intent and distinguish them using entities. In this example, #open-account could be the intent and @credit and @debit could be entities.

Entities

Any entities that you identified as particularly problematic during the Analyze phase should be investigated further and improved. The first step should be using the IBM Watson Assistant Entity Expansion Recommender to expand the synonyms for the values within that entity. The recommender finds related synonyms for you based on contextual similarity extracted from a vast body of existing information (including large sources of written text), and uses natural language processing techniques to identify words similar to the existing synonyms in your entity value. This could include pure synonyms, but also slang, abbreviations, and common misspellings. Manual expansion of the entity should also be performed as appropriate, including additional values, or synonyms not suggested by the recommender.

Dialog

For user messages that were wrong due to incorrect dialog logic, the relevant dialog nodes should be inspected and improved. This means you might need to re-arrange dialog logic or add to existing logic.

Analyze and Improve – Coverage

The analysis of Coverage can be broken down into two categories:

(1) An intent was found. For these utterances, an intent was found even though no answer was given. These utterances need to be assessed so you can identify whether the problem was because an entity was missed (i.e. not extracted from an utterance), or whether the dialog logic needs to be extended. This can be performed with the same procedure described in the Effectiveness section.

(2) No intent was found. These utterances had no response because an intent was not classified to it above the confidence threshold. Poor coverage due to missing intents requires expanding your intent training, possibly creating new intents in the process. This is the focus of the remainder of this section.

Applying Disambiguation

The “Create Training for Initial Deployment” phase suggests that you rely on disambiguation to help your users clarify their meaning when the assistant isn’t sure what the user wants. If your skill is not providing disambiguation options, verify that the disambiguation feature is enabled in your Skill settings and review your dialog node names for actions that you want to include as disambiguation choices. Use of disambiguation will usually improve your coverage since it will often provide clarification choices to the user instead of giving no response.

Using Intent Coverage Expansion Recommendations

The traditional approach to expanding intent coverage is performed by manually examining utterances from the production logs, looking for patterns your current intents are not covering and shaping them into your training. The good news is that Watson Assistant includes a recommendation tool that does the heavy lifting for you. The same Intent Recommendations feature introduced in the Create Training for Initial Deployment phase also analyzes chat transcript logs generated by your assistant’s user conversations to find candidate intents and user examples that can expand your coverage.

Recommendations for new intents are based on techniques of unsupervised learning to group related user utterances that were classified with low confidence. Iterating by viewing and acting on these recommendations will expand the “breadth” of your coverage to more topics. The Recommender for new user examples feature follows a principle of “more like this,” helping you find additional ways of expressing the intents you already have. You might think of this as expanding the “depth” of your coverage.

Manually Identifying Areas of Expansion

If you'd like to manually expand your intent coverage instead of using Watson Assistant recommendations, you can examine utterances from your assistant production logs, focusing on those utterances that are below the confidence threshold you have set for your intents (0.2 by default). These are the utterances that your current intents are not covering, and thus are excellent candidates to include.

You can optionally narrow down your search by looking at log entries that are below *but close to* the confidence threshold, particularly if you have raised your threshold significantly higher than 0.2. This will focus your expansion on topics that are "closer" to your existing training and may lead to candidate utterances for improving the depth of your existing intents. Examining utterances with confidence below the confidence threshold will cover both range and depth of coverage, ensuring you discover completely new intents as well.

The creation of intents and user examples can be prioritized based on the patterns of utterances that occurred most frequently during the assessment.

Intent Coverage – Incrementally adding new intents

When expanding coverage, you probably discovered that entities, or dialog logic are also needed. If you'd like to gain feedback on which new intents are needed most before going through the full process of adding them to your application, you can create *silent intents*. Silent intents are intents that you're considering adding to your system, and they're created and trained just like any other intent. However, when these intents match, you do not produce a response visible to the user and still output your standard response for, "I'm sorry, I'm not trained on that."

You can monitor the frequency of these intents matching in the production logs in Watson Assistant's Improve tab, and if there is sufficient volume matching these intents, you can fully enable them at a later date by adding the corresponding training examples, entities, and dialog logic.

Analyze Training and Test Data

After you've made changes, it's ideal to verify that the new version is performing well before you release it. As shown in Figure 1, the Continuous Improve Process includes iteratively testing and analyzing your training and test data, If your test results do not meet your expectations, you can use the Dialog Skill Analysis notebook to help identify problems with your training or test data before you deploy. This notebook includes a variety of data analysis and model analysis capabilities:

- Training data characteristics and class imbalance
- Terms correlated with intents
- Confidence threshold visualizations
- Detailed test performance metrics

Pre-Deploy Testing

Testing is vital to prevent regressions before you release an update. This can be accomplished by creating a pre-deploy *test set.* A test set is a collection of user messages paired with the expected system behavior.

For example, a test set might contain the user message, "How do I find the library" and the expected system behavior:

- Intent: #locate
- Entity: @library

- System Response: “The library is located at …”

Test sets can serve two purposes:

(1) Catch major regressions

Testing helps catch mistakes or regressions in the new version, like uploading the wrong workspace for release, or forgetting to merge in some of the intents. This purpose places fewer requirements on test set creation. Test sets could be created ad-hoc, such as a collection of important examples that you would like to test for and check on each release. If they fail, you may want to delay the release. This is analogous to unit testing in software engineering.

(2) Quality metrics

If they’re created and maintained properly, test sets can track quality metrics of your system over time and help you understand whether your changes produced a positive impact. This could include metrics like precision, recall, and coverage - all of which can be calculated using the Measure and Analyze Notebooks. However, more care is required in the creation and maintenance of the test set to ensure the metrics reported are meaningful:

a. The test set must be a random sampling of your overall production logs so that the test set metrics represent the overall experience as seen by your assistant users.
b. You should not use logs that were included in the Analyze phase, which may have been pulled from a subset of conversations, such as escalated conversations. You must not add an utterance from the log to both your training data and your test set.

Test Set Management

How a test set is created determines which of those scenarios it can be used for. This section will describe creating and managing test sets.

Creating a Test Set

Creating an initial test set for tracking performance is simple. User messages should be randomly sampled from the production logs, labeled with the expected behavior, and stored as a test set. Your test set should not contain utterances that are also in your intent training.

Test Set Size

Test set size determines the confidence intervals on the metrics produced. The process of computing confidence intervals for test sets is the same as for computing assessment results, as described in Appendix A. The Analyze Effectiveness Notebook computes the confidence interval of your test set metrics for you automatically. These intervals should always be used when comparing changes in test results, along with the understanding that changes are not statistically significant if the confidence intervals overlap. Confidence intervals can be tightened by increasing the size of your test set.

It is important to note that the confidence intervals are only accurate if your test set is a random sample from your production logs. If your test set is skewed (either because it was not random from the start or because your production log traffic has changed over time), then this skew will further increase the potential error, effectively invalidating (and widening) the confidence intervals. Therefore, it is important to keep your test set up to date with the current log data.

Keeping a Test Set Up to Date

When a test set is used to measure performance, it is critical to ensure the test set is an accurate reflection of your production log and workspace data. There are three common strategies for doing this:

(1) Test set rotation. The test set can be continuously updated with new, randomly sampled log entries to keep it up to date with production logs. When they expire, test entries are annotated with the date they were added and dropped from the test set.

(2) Distribution analysis. Ensure that distribution of intents and entitles in production match the distribution in your test set.

(3) Accurate labels. As you change the definition of intents and entities you need ensure that the labels in the test set are kept up to date. For example, if you combine two intents, utterances in the test set that previously were labeled with one of the combined intents need to be updated to be labeled with the combined intent name.

A test set's randomness can be evaluated by comparing the distribution of key characteristics to the ones in the production logs. For example:

- Intent distribution
- Utterance length distribution
- Channel breakdown (mobile versus web)
- User type (such as new versus existing and demographic)

If any of these distributions show significant differences beyond what you'd expect from random sampling, then it suggests that the test set is no longer a good representation of the overall production traffic. But you can take additional examples from the production log that have the characteristics you're lacking, and add them to the test set so you can bring the distributions more in line.

Deploy

Once you have conducted a test, have confidence you've improved your assistant, and caused no regression, you should deploy the new version of your assistant to production. After deployment, you will be in the Measure Live System phase again. From here, you can monitor your metrics to further verify your improvements and renew the cycle again as you continue to improve your Watson Assistant.

Appendix A. Selecting Sample Sizes

Manual assessments and test sets both require labeling of log data, which requires human effort. Thus, they are both performed on a sample, or subset, of total log data over a period of time. When a summary metric is computed from sampled data, the summary metric should always be reported together with a *confidence interval.* The confidence interval shows how much different your sample is from the total log data of the time period the sample was taken from.

Example: Reporting that precision=0.75 is meaningless without confidence interval to show how likely the precision of the sample represents the precision of the total logs. It could have been computed from a sample size of four messages, in which case it is highly unlikely an accurate representation of the precision of total logs.

Instead, you should report that precision=0.75 with a 95% confidence interval of 0.70-0.80. This tells you that overall precision has a 95% chance of being 0.70-0.80, and a 5% chance of being below 0.70 and above 0.80. More simply, there's a 95% chance your precisions is between 0.70-0.80.

As the size of the sample (N) increases, the confidence interval tightens. The tighter the interval, the more confident you can be that the sample represents the total. We recommend targeting a confidence interval with an error margin between 0.1 and 0.05, which means your sample size should be between 100 and 400 messages**.**

To get the confidence interval, apply the formula 1/sqrt(N), where N is your sample size. The output of the formula will be your error margin, and your confidence interval will be precision plus and minus the error margin. The above example had an error margin of .05. Our Analyze Effectiveness Notebook will calculate confidence intervals for you and the below table gives some examples of confidence intervals and error margin given different sample sizes. Note that the size of your sample is the only variable that impacts the confidence interval, and the size of the total logs has no impact.

N	Error	95% Confidence Interval for Precision = 0.7
100	0.1	**[0.6 – 0.8]**
400	0.05	**[0.65 – 0.75]**
1000	0.03	**[0.67 – 0.73]**

Appendix B. Notebooks for Continuous Improvement

Watson Assistant provides four Jupyter notebooks to assist you in adopting the continuous improvement best practices. These notebooks have been described throughout this document. This Figure summarizes where these notebooks fit along the phases of the lifecycle. These Jupyter notebooks are available for use in Watson Studio or another Jupyter environment of your choice. Please see the Watson Assistant github repository to get started.

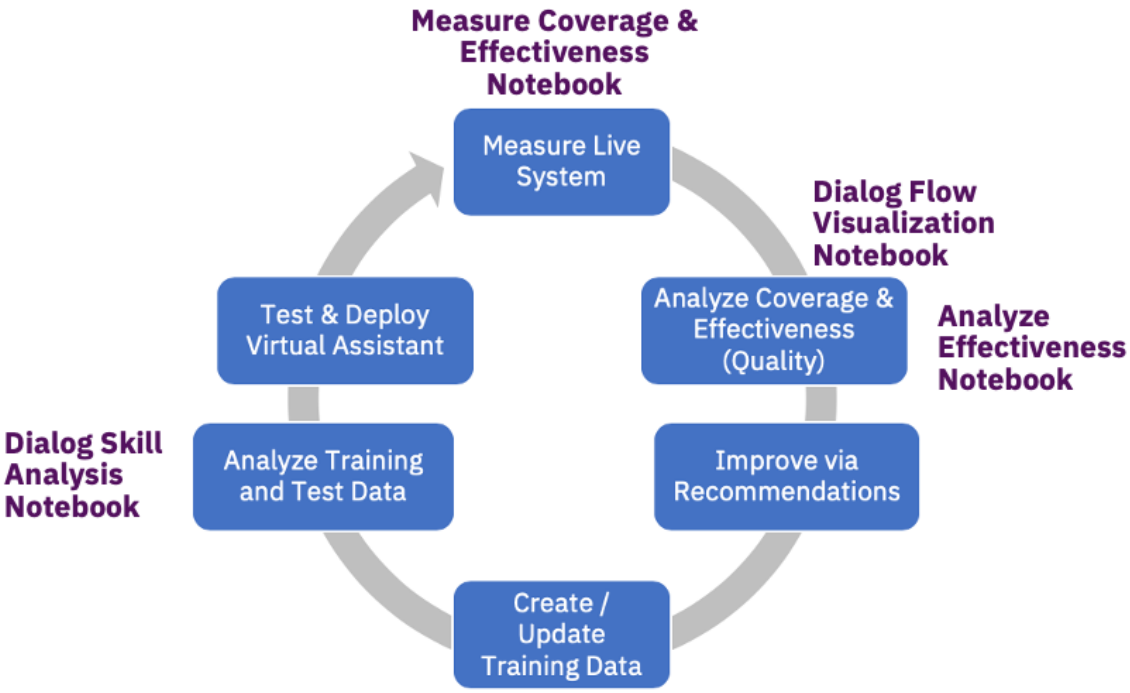

Figure 9. Notebooks for Improvement Phases