Watson Assistant Continuous Improvement Best Practices

Introduction

This document details best practices for the on-going management of a Watson Assistant based virtual assistant after it is deployed to production. It defines steps and concepts integral to effectively and continuously improving your virtual assistant. The recommendations in this document come from experts in the virtual assistant space and are based on a large number of client experiences.

Business KPIs are King

No matter how engaging or accurate your virtual assistant is, it should be making a positive impact on your business. There are many potential impacts which the virtual assistant could be expected to achieve and those identified should be aligned to the individual business.

The specific business outcome should be identified, and specific metrics identified to measure it. For example, if your business goal is to reduce customer service cost, 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. 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 the business and customer objectives it is imperative that there is a formal process which is executed.

An effective improve process will meet the below criteria:

(1) Provides a reliable understanding of overall performance of the virtual assistant
(2) Clearly shows where you should be prioritizing your improvement effort
(3) Allows you to make the necessary improvements as efficiently as possible

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

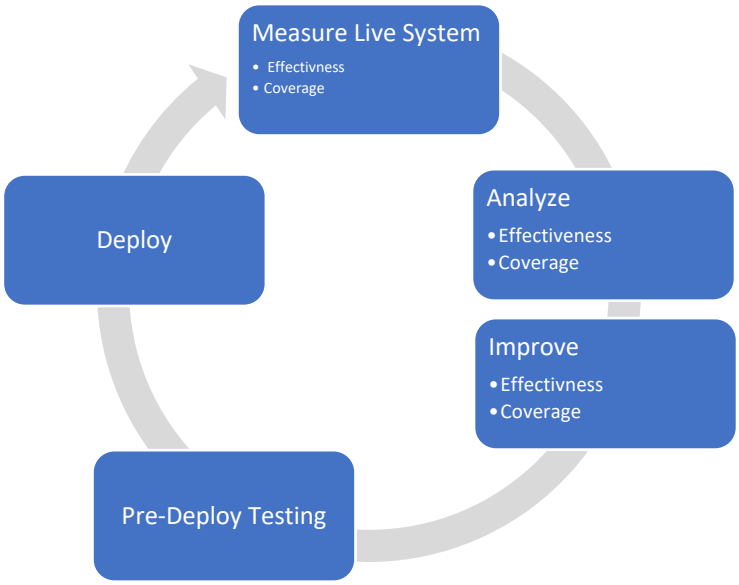

Figure 1: Continuous Improvement Process

Resources for Continuous Improvement

The Watson Assistant service provides a variety of features to help you continuously improve your assistant. These features include improvement recommendations from Watson and analytics to improve based 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 they should be used for continuous improvement.

Measure Live System

The goal of this stage is to understand where your virtual assistant is doing well vs where it is not. The first step of your continuous improvement process is to be able to monitor and understand the behavior of your system, often done through automated metrics.

Effectiveness and coverage are the two measures that provide a reliable understanding of the overall performance of your assistant. The combination of effectiveness and coverage 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 coverage or coverage are low, the metrics provide you with the information you need to start improving your assistant.

Effectiveness

The first measure which should be the focus of your improvement effort is "Effectiveness" at the conversation-level. Effectiveness measures how well your assistant is handling the conversations or messages it is responding to. This includes any automated measures you may have that help identify problematic conversations. These include metrics such as:

Conversation Containment

Conversation containment is the portion of conversations not handed off or escalated to a human agent for quality reasons. Assuming one of your assistant's objectives is to handle conversations otherwise handled by humans, if your assistant is unable to address a customer's need in a conversation and hands the conversation to a human, 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 specify they want one. Any time a conversation hits a connect to human agent response type or intent, the conversation has not been contained.

Tasks Completion

Task completion measures whether or not your assistant completed the task asked of it by the customer during a conversation, regardless of whether or not the conversation was contained. Tasks can be defined as either transactional or informational. A transactional task refers to helping a customer through a process during a multi-turn conversation (i.e. booking an appointment). An informational task refers to simply providing requested information. You can measure transactional tasks by labeling dialog nodes that mark the completion of a task. If an intent trained to initiate a task was hit, but the completion node was not, the conversation can be deemed ineffective. You can measure informational tasks by prompting customers to express whether or a not the information was helpful after it is provided. Conversations where a customer receives information but marks it as not helpful 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.

You can implement all of these measures using the Watson Assistant Measure Notebook.

Coverage

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

Coverage is measured as a percentage of total messages for which the assistant returns a real response. You can measure coverage leveraging intent confidence thresholds and anything_else dialog nodes. If a message does not hit the default intents 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 assistant should or should not cover is a critical decision you and your team should make. For example, if you are creating an assistant to assist customers in getting their credit score, you should not spend time training your assistant on how to answer question on credit card interest rates. However, if you see that customers are frequently asking about certain topics your assistant is not defined to cover, you should train it to respond to these topics in a way that communicates its coverage scope to the customer. Continuing with the interest rate example, you could train your assistant to respond generally to such inquiries with "I understand you are requesting interest rate information, unfortunately I cannot handle questions on this topic. For interest rate information please follow the link below." Messages triggering these responses should be considered covered.

Effectiveness & Coverage

Analysis of effectiveness and coverage should be complementary. If your assistant is answering questions incorrectly, you improve this by focusing on improving effectiveness. If your assistant is not answering enough questions, you improve this by focusing on expanding coverage. For a given effectiveness metric (say containment) you can view coverage separately for both the escalated and not-escalated conversations to provide more insight into where you might focus first.

For example, consider the Containment and Coverage metrics as reported in Figure 2 which comes from the Watson Assistant Measure Notebook, where containment is our measure of effectiveness. In this scenario, 73% of conversations are escalating to a human agent, so there are many conversations that are not being contained. The assistant is responding to a significant number of questions in both the contained (84%) and not-contained (88%) categories, so this suggests the quality of the answers in the not-contained conversations is likely the problem. Moreover, the average intent confidence is significantly lower for not-contained conversations (53% vs 85%), increasing the concern over response quality. Together this suggests that focusing on the *covered questions* from *the not-contained conversations* is a promising place to start your investigation. In other words, focus on improving effectiveness.

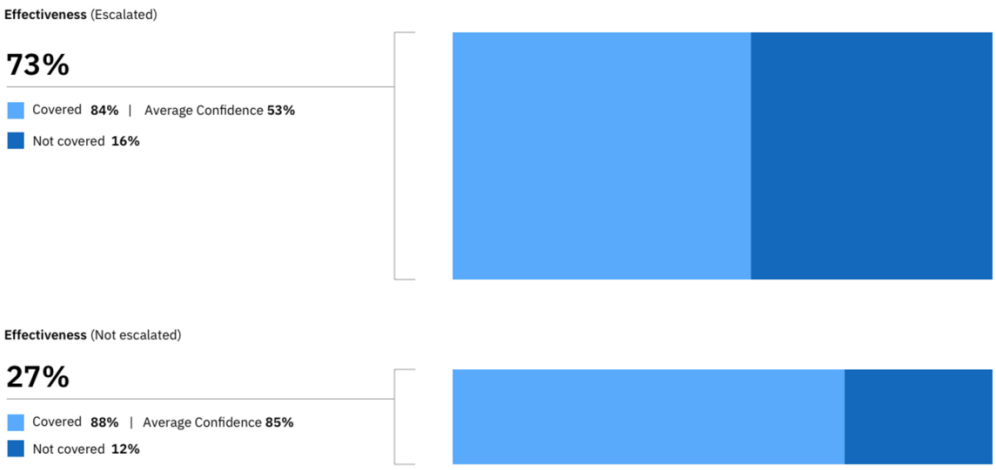

Figure 2: Effectiveness & Coverage

The Watson Assistant Measure Notebook helps you narrow your immediate focus of improvement by looking at effectiveness and coverage analytics. You can also choose to improve both the effectiveness *and* coverage dimensions, but they are most often improved as separate processes. The immediate next sections outline steps toward analyzing and then improving effectiveness. Following that, we outline steps for analyzing and improving coverage.

Analyze – Effectiveness

If you have decided to analyze your assistant's effectiveness, you will want to perform a deeper analysis on the group of ineffective conversations you identified (i.e. the not-contained conversations) from the Measure Live System phase. All of the following practices can be achieved using IBM's Analyze Effectiveness Notebook.

This analysis is performed by:

(1) Sampling a random set of utterance-response pairs from ineffective conversations you are analyzing.
(2) Performing a manual assessment of those sampled logs.
(3) Analyzing the results.

Sampling

Question-response pairs should be sampled randomly from the group of ineffective conversations you chose to investigate in the Measure Live System phase. This could be all conversations in your log, or a subset of conversations marked as ineffective, such as not-contained conversations. If, for example, you are only analyzing not-contained conversations, then you would randomly sample utterance-response pairs from the not-contained conversations. If you are analyzing all conversations, then you would sample utterance-response pairs from all conversations.

Appendix A - Selecting Sample Sizes provides guidance on the number of conversations to sample for a test set.

Performing a Manual Assessment

A manual assessment is performed as follows.

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 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”

Perform Analysis

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

Summary Metrics

Our Analyze Effectiveness Jupyter 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). These intervals should always be used when comparing assessment results, with the understanding that changes are not statistically significant if the confidence intervals overlap. Confidence intervals can be reduced by increasing the size of your sample.

Intents

Intent analysis is performed by looking for the dominant patterns of intent errors. An 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 (i.e. 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 when they should not be, thus hiding the correct intent(s). These intents likely have training examples that clash with 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. Figure 3 is an examples of confused intents pairs and was taken from IBM's Analyze Effectiveness Notebook.

View the whole list here: **ConfusedIntentPairs.csv**

25 Worst Intents

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 3: Confused Intent Pairs

Entities

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

Dialog

IBM's Analyze Effectiveness Notebook will also highlight instances where dialog problems were the root cause of the ineffectiveness. Dialog could be the problem because either the wrong response condition was triggered or because there was no response condition in place to handle the user message. These situations are used as candidates for improvement.

Improve – Effectiveness

The first 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, the user message can be added 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.

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 a 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 identifies these user examples for you to resolve. This feature helps you find training examples that may have been added to the wrong intent and are confusing your classifier. If you see a significant number of conflicts between intents and you are unable to resolve them, your intents may be too nuanced and you should consider combining them into one intent, as described further below in Section *Combining Intents.*

Adding Training to Clarify Boundaries

Once conflicts between confused intents have been fixed, the next best thing to do is to add training designed to help clarify the boundary between intents. In other words, add training to

help 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 be clearer.

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), training should be added to the intents that are being overshadowed by the problematic 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 are training with a high confidence (i.e. above 0.80). The intent is clearly not struggling with these examples. It is better to add examples that were matched at a low confidence, so the intents will correctly classify similar utterances in the future that it previously did not.

Combining Intents and Using Entities

If you have intents that you are struggling to differentiate using the approaches above, you should evaluate whether the intents are too narrow or nuanced, and 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 were 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 within Watson Assistant to expand the synonyms for the values in that entity. The recommender finds related synonyms 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, common misspellings, etc. 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 – 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 to 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.

Identifying Areas of Expansion

Expanding intent coverage is performed by examining utterances from the 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 candidates for including.

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.

Once you have narrowed your utterances to focus on intent expansion, you can analyze in one of two ways:

(1) One-by-one: This is similar to the assessments performed in the Effectiveness section. Each utterance is examined one by one to determine (a) what the intent should have been, and (b) whether a new intent needs to be created

(2) Unsupervised learning: Utterances can be clustered to show groups of semantically similar utterances that are candidates for (a) adding to training of an existing intent, or (b) creating a new intent. This is the same principle as (1) above but is potentially more efficient by operating on groups of utterances that were classified with very low confidence. Unfortunately, at this point in time we do not have any clustering tools of this type to provide, however, Watson Studio does provide guidance on clustering and a SPSS analytics algorithm to assist.

Improve - Coverage

The first action to improve your assistant's coverage is to update the training of any existing intents or entities based on the results of the assessment performed in the Analyze phase. This is similar to the steps performed when improving effectiveness.

However, when expanding coverage there is a good chance that you also discovered that new intents, entities, or dialog logic are needed.

Consider creating new intents based on the utterances that were missing intents during the coverage assessment. The creation of intents can be prioritized based on those that occurred most frequently during the assessment.

If you would 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 are considering adding to your system. They are created and trained just like any other intent; however, when these intents match you do not produce a response visible to the user – you still output your standard response for, "I'm sorry, I'm not trained on that". However, 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, fully enable them at a later date by adding the corresponding training examples, entities and dialog logic.

Pre-Deploy Testing

After you have made changes to your skill it is desirable to verify that that the new version is performing well before you release it. 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. For example, if you uploaded the wrong workspace for release, or forgot 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 choose to delay the release. This is analogous to unit testing in software engineering

(2) Quality metrics

If created and maintained properly, a test set can help you track quality metrics of your system over time and help you understand whether your changes produced a positive impact. This could include metrics such as precision, recall and coverage which can be

measured using our measure and analyze notebooks. More care is required in the creation and maintenance of the test set to ensure the metrics reported are meaningful:

- 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.
- 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 utterances from the log to both your training data and your test set

After your updated assistant has passed a test against your test set, you are ready to deploy this new version to production. Once in production, you should monitor the metrics mentioned in the above measure section to fully understand if your efforts have made improvements.

Test Set Management

How a test set is created determines which of those scenarios it can be used for.

Creating a Test Set

Creating an initial test set for tracking performance is straightforward. 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 you 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, described in Appendix A. Our 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, 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 accurate *only* 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 (by 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 two 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. Test entries are annotated with the date they were added and dropped from the test set when they expire.

(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 that of the production logs. For example:

- Intent distribution
- Utterance length distribution
- Channel breakdown (mobile vs web)
- User type (new vs existing, demographic, etc)

If any of these distributions show significant differences (beyond what one would expect from random sampling) it suggests that the test set is no longer a good representation of the overall production traffic. You can sample, and add to test set, additional examples from the production log that have the characteristics you are lacking, to bring the distributions more in line.

Deploy

Once you have conducted a test and are confident you have improved your assistant and caused no regression, you should deploy the new version of your assistant to production. After you deploy you will be in the Measure Live System phase again. From here you can monitor your metrics to further verify your improvements and begin the cycle anew as you iterate once more to improve your assistant even more.

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, and 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 your confidence interval you 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, 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]**