Conversational AI Platform Comparison Study
Contents

Which Conversational AI Platform Is Best for Your Business? ........................................... 3

Study Process ................................................................................................................................ 4

Conversational NLU Performance Quantitative Evaluation .................................................. 8

Qualitative Analysis of Each Platform ...................................................................................... 13

Summary ........................................................................................................................................... 31
Conversational AI is on the rise. As Natural Language Processing (NLP) capabilities improve, the applications for conversational AI platforms are growing. One popular application entails using chatbots or virtual agents to let users request the information and answers they seek.

A growing number of businesses offer a chatbot or virtual agent platform, but it can be daunting to identify which conversational AI vendor will work best for your unique needs. We studied five leading conversational AI platforms and created a comparison analysis of their Natural Language Understanding (NLU), features, and ease of use.

Our analysis should help inform your decision of which platform is best for your specific use case.

- Study Process
- Conversational NLU Performance Quantitative Evaluation
- Qualitative Analysis of Each Platform
- Summary
To conduct this study, we selected the following five platforms based on popularity and ease of access:

- AWS Lex
- Google Dialogflow CX
- IBM Watson Assistant
- Kore.ai
- Microsoft LUIS

Each platform is listed in alphabetical order throughout our study.

Other highly competitive platforms exist, and their exclusion from this study doesn’t mean they aren’t competitive with the platforms we reviewed.

During the course of this study, we set up a series of tests and evaluations to perform on each of the five platforms. We looked at them across two primary dimensions:

1. Quantitative analysis of NLU performance
2. Qualitative analysis focused on ease of use, technical capabilities, and business considerations

We actively used each platform to build models for three different intents in three industries: healthcare, finance, and retail. The specific intents per category were as follows:

**Healthcare**

1. Find care
2. View claim status
3. Check benefits
Study Process continued

Finance
1. Check balance
2. Make payment
3. Create account

Retail
1. Locate a store
2. Make a return
3. Order status

To gather a variety of potential phrases — or “utterances” — for use in training and testing each platform, we submitted utterances that consumers could potentially use for each of these intents. We selected 50 utterances for each intent. Fifteen utterances were also created for the "None" intent in order to provide the platforms with examples of non-matches.

Once the corpus of utterances was created, we randomly selected our training and test sets to remove any training bias that might occur if a human made these selections. The five platforms were then trained using the same set of training utterances to ensure a consistent and fair test.

To help us learn about each product’s web interface and ensure each service was tested consistently, we used the web interfaces to input the utterances and the APIs to run the tests.

Some of the services maintain thresholds that won’t report a match, even if the service believed there was one. They do this when the confidence level is relatively low. However, to treat each service consistently, we removed these thresholds during our tests.
The study data was obtained using the API interface of each service to create three bots (one per category). We then performed three iterations of training and testing.

In each iteration, the models were trained with a specific number of training utterances and tested with all 25 of our test utterances that were not used for training in any of the three rounds. The iterations were structured as follows:

- **Iteration one:** 15 training utterances added to each intent, and 4 training utterances added to the "None" intent
- **Iteration two:** 5 more training utterances added to each intent, and 2 more training utterances to the "None" intent
- **Iteration three:** 5 more training utterances added to each intent, and 2 more training utterances to the "None" intent

Next, an API integration was used to query each bot with the test set of utterances for each intent in that category. Each API would respond with its best matching intent (or nothing if it had no reasonable matches). All default confidence thresholds were removed to level the playing field. To evaluate, we used Precision, Recall, and F1 to qualify each service's performance.

Throughout the process, we took detailed notes and evaluated what it was like to work with each of the tools. We also performed web research to collect additional details, such as pricing.
We compared five major conversational AI platforms based on their NLU performance, features, and ease of use.
Conversational NLU Performance Quantitative Evaluation

This section quantitatively outlines the NLU performance of each platform. As you review the results, remember that our testing was conducted with a limited number of utterances. All platforms may perform better when provided with more data and any tool-based advanced configuration settings.

The three major metrics we collected were Precision, Recall, and the F1 score. These are defined as follows:

- **Precision** - Total number of correct matches reported/total number of matches reported. In plain English, what percentage of the matches reported are actually matches?

- **Recall** - Total number of correct matches reported/total number of expected matches. In plain English, what percentage of the matches we wanted the platform to detect did it actually detect?

- **F1 score** - \(2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\). In plain English, this is the harmonic mean of the Precision and Recall Scores.

The F1 score was used as an overall rating of each platform’s performance. While other metrics for evaluating performance can be used, we chose F1 because it is a well-known standard in the industry, and the reasons for choosing a certain measurement method is best done with an understanding of the specific application that your NLU algorithm is being used for.

We ran multiple rounds of tests for each category and intent. This report includes the scores based on the average round three scores for each category. The score values run from 0 to 1, with 1 being the highest score.
In this category, Watson Assistant edges out AWS Lex for the best net F1 score, but the gap between all five platforms is relatively small.
We see a similar story when we look at the Finance category:

**Final Scoring for the Three Intents**

*Finance Category*

<table>
<thead>
<tr>
<th>Platform</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Lex</td>
<td>1.0</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>Google Dialogflow CX</td>
<td>0.9</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>IBM Watson Assistant</td>
<td>0.8</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>Kore.ai</td>
<td>0.7</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>Microsoft LUIS</td>
<td>0.6</td>
<td>0.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Platform Ranking Based on F1 Score**

*Finance Category*

<table>
<thead>
<tr>
<th>Virtual Agent Vendor</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Lex</td>
<td>2</td>
</tr>
<tr>
<td>Google Dialogflow CX</td>
<td>4</td>
</tr>
<tr>
<td>IBM Watson Assistant</td>
<td>1</td>
</tr>
<tr>
<td>Kore.ai</td>
<td>3</td>
</tr>
<tr>
<td>Microsoft LUIS</td>
<td>5</td>
</tr>
</tbody>
</table>
Finally, the category where we saw the most variance was Healthcare:

**Final Scoring for the Three Intents**

Healthcare Category

As previously noted, each platform can be trained across each of the categories to obtain stronger results with more training utterances.
IBM Watson Assistant received the highest F1 Scores in the Retail and Finance categories, while AWS Lex took first place in the Healthcare category.
Qualitative Analysis of Each Platform

When designing this study, we wanted to evaluate each platform both quantitatively and qualitatively. In addition to understanding the NLU performance and amount of training data required to achieve acceptable confidence levels, we wanted to know how easy it is to enter training utterances, test intents, and navigate each platform.

Here is our assessment of each platform from a technical and non-technical perspective.

AWS Lex
We performed our test with the AWS Lex V1 console. The AWS Lex V2 console has been released and is very similar, but it does improve upon a few things. We’ll aim to mention these improvements as they come up, but it’s worth noting this as a blanket point.
Qualitative Analysis of Each Platform continued

Overall Platform Impressions
AWS Lex has a simple, table-based approach. The pages aren’t surprising or confusing, and the buttons and links are in plain view, which makes for a smooth user flow.

The look and feel are homogeneous with the rest of the AWS platform — it isn’t stylish, but it’s efficient and easy to use. Experienced AWS Lex users will feel at home, and a newcomer probably wouldn’t have much trouble, either.

The graphical interface AWS Lex provides is great for setting up intents and entities and performing basic configuration. AWS Lambda is required to orchestrate the dialog, which could increase the level of effort and be a consideration for larger-scale implementations.

Advantages
AWS Lex provides an easy-to-use graphical interface for creating intents and entities to support the dialog orchestration. The interface also supports slot filling configuration to ensure the necessary information has been collected during the conversation. AWS Lex is one of the least expensive virtual agent options we evaluated.

AWS Lex supports integrations to various messaging channels, such as Facebook, Kik, Slack, and Twilio. Within the AWS ecosystem, AWS Lex integrates well with AWS Kendra for supporting long-tail searching and AWS Connect for enabling a cloud-based contact center.
Qualitative Analysis of Each Platform

Downsides
Although the interface is available for basic configuration, AWS Lambda functions must be developed to orchestrate the flow of the dialog. Custom development is required to use AWS Lex, which could lead to scalability concerns for larger and more complex implementations.

Setup, Training, and Testing
Setting up a virtual agent is lean and straightforward. AWS Lex provides samples that can help a beginner get a feel for things, so starting from scratch isn’t confusing.

When entering training utterances, AWS Lex was the only platform where we had issues with smart quotes — every other service would convert these to regular quotes and move on. Also, the text input fields can behave strangely — some take two clicks to be fully focused, and some place the cursor before the text if you don’t click directly on it. This quirk doesn’t appear to be a problem in the V2 console. The entry process is otherwise smooth and responsive.

You can choose to return all API information in the AWS interface or receive summary information when testing intents. The offering is standard and simple to follow. All chat features are tightly packed to the right side of the screen, making it easy to work intently.

The AWS API offers libraries in a handful of popular languages and is the only platform that provides a PHP library to directly work with Lex. Developers may have an easier time integrating with AWS services in their language of choice, taking a lot of friction out of a project — a huge plus.
Qualitative Analysis of Each Platform continued

Business Considerations

Roadmap
AWS Lex appears to be focused on expanding its multi-language support and infrastructure/integration enhancements. There seems to be a slower pace of core functionality enhancements compared to other services in the space.

Cost Structure
AWS Lex is priced using two models: request and response interaction, and streaming conversation.

• In a request and response scenario, each user input is processed as a separate API call
• In a streaming conversation scenario, all user inputs across multiple conversation turns are processed in one streaming API call
Qualitative Analysis of Each Platform continued

The request and response pricing is $0.004 per speech request and $0.00075 per text request. The streaming conversation pricing is $0.0065 per 15-second interval for streaming speech and $0.002 per request for streaming text input.

Google Dialogflow CX

Overall Platform Impressions
Google Dialogflow is clean and straightforward. The “flows” section can seem a bit daunting, but it works elegantly once you get going.

The look and feel is second only to Kore.ai. The main section of the interface moves and flows around the different panels being used, making it a joy to use.
Advantages
Google Dialogflow provides a user-friendly graphical interface for developing intents, entities, and dialog orchestration. Within the Dialogflow, context setting is available to ensure all required information progresses through the dialog. Webhooks can be used for fulfillment within the dialog to execute specific business logic or interact with external applications.

Google Dialogflow offers a range of integrations with multiple messaging channels. A notable integration is the ability to utilize Google’s Phone Gateway to register a phone number and quickly and seamlessly transform a text-based virtual agent to a voice-supported virtual agent.

Downsides
Some challenges exist when working with the dialog orchestration in Google Dialogflow ES. Those issues are addressed in Google Dialogflow CX, which provides an intuitive drag-and-drop visual designer and individual flows, so multiple team members can work in parallel. The new version of Google Dialogflow introduces significant improvements that reduce the level of effort required for a larger-scale virtual agent, but it comes at a significantly higher cost.

Setup, Training, and Testing
Setting up an agent is relatively simple. There are several options, but if you move slowly and read everything, it isn’t a hassle.

Entering training utterances is easy and on par with the other services, although Google Dialogflow lets you supply a file of utterances.
Qualitative Analysis of Each Platform  continued

In its interface, Google Dialogflow CX focuses heavily on controlling the conversation’s “flow.” Google also provides their API data in the interface chat function. Much of the data has to do with conversational context and flow control, which works wonders for people developing apps with long conversational requirements.

We had some difficulty setting up the API. For example, all the data needed to piece together an API endpoint is there, but it would be nice to see it auto generated and presented to the user like many of the other services do.

Business Considerations

Roadmap
Google Dialogflow has been rapidly rolling out new features and enhancements. The recent release of Google Dialogflow CX appears to address several pain points present in the Google Dialogflow ES version. It appears Google will continue to enhance and expand on the functionality the new Google Dialogflow CX provides.

Cost Structure
Google Dialogflow CX is priced on a per request/minute basis. A request is any call to the Google Dialogflow service, whether direct with API usage or indirect with integration or console usage.

- Text interactions cost $0.007 per request
- Voice sessions cost $0.06 per minute

Each voice session is charged $0.001 per second of audio, with a minimum of one minute.
Overall Platform Impressions
The IBM Watson Assistant interface is easy to navigate after a few minutes of exploring. Some parts of the interface are initially vague but become straightforward once you understand the structure and components.

Overall, the interface looks and feels snappy. IBM Watson Assistant lazy loads most elements, but the extra startup time is hardly noticeable, which is commendable.
Advantages
IBM Watson Assistant provides a well-designed user interface for both training intents and entities and orchestrating the dialog. The point-and-click interface enables you to get up and running quickly.

The product supports many features, such as slot filling, dialog digressions, and OOTB spelling corrections to create a robust virtual agent. Webhooks can be used within the dialog nodes to communicate to an external application based on conditions set within the dialog.

IBM Watson Assistant supports integrations to various SMS and IVR providers. Webhooks can be utilized within dialog nodes to interact with external services to extend the virtual agent’s capabilities. IBM Watson Assistant can integrate with IBM Watson Discovery, which is useful for long-tail searching against unstructured documents or FAQs.

Downsides
Although a robust set of functionalities is available, IBM Watson Assistant is one of the more expensive virtual agent services evaluated.

Setup, Training, and Testing
Setting up an agent is heavy with platform-specific jargon, which can cause some challenges. Navigation is quick enough that the system feels slicker than it is.
Qualitative Analysis of Each Platform continued

When entering training utterances, IBM Watson Assistant uses some full-page modals that feel like a new page. This made us hit the back button and leave the intent setup completely, which was a point of frustration. Aside from that, the interface works smoothly once you know where you are going.

IBM Watson Assistant’s testing interface is robust for both validating the intent detection and the flow of the dialog. When interacting with the test interface, IBM Watson Assistant provides the top-three intent scores and the ability to re-classify a misclassified utterance on the fly. By clicking on the responses, the specific nodes of the dialog are highlighted to show where you are in the conversation — this helps troubleshoot any flow issues when developing more complex dialog implementations.

In terms of testing intents in the API, the publishing flow was standard, straightforward, and clear. The API endpoint publishing isn’t as easy as Microsoft LUIS, but it isn’t far off.

Business Considerations

Roadmap
IBM Watson Assistant has a robust base of features and functionality that IBM appears to be continuously improving.

Cost Structure
IBM Watson Assistant follows a Monthly Active User (MAU) subscription model. The plus plan is $140/month and includes up to 1,000 monthly active users. Any additional MAUs are $14 per 100 MAUs. Enterprise licensing can be negotiated with IBM.
Qualitative Analysis of Each Platform continued

**Kore.ai**

**Overall Platform Impressions**
Kore.ai is a joy for new users. It provides a walkthrough feature that asks for your level of NLP expertise and suggests actions and highlights buttons based on your response. This enables users to get up and running in a few minutes, even if they’ve never seen the site before.

**Advantages**
Kore.ai lets users break the dialog development into multiple smaller tasks that can be worked on individually and integrated together. It also supports the ability to create forms and visualizations to be utilized within interactions. Knowledge graphs are supported for integrating question and answer functionality.

Kore.ai offers a range of integrations with multiple messaging channels and is the winner in the look and feel department. The app is easier on the eyes than the others without sacrificing efficiency.
Qualitative Analysis of Each Platform  continued

Downsides
Kore.ai provides a single interface for all complex virtual agent development needs. There are many configuration options across NLU, dialog building, and objects within the channel. Given the amount of features and functionality available to develop and refine complex virtual agents, there is a learning curve to understand all the offerings.

Setup, Training, and Testing
Kore.ai provides a robust user interface for creating intent, entities, and dialog orchestration. Within the interface, it offers a significant number of features for handling complex functionality.

Kore.ai’s input fields move when you select them, so one small issue was that it was a little slow entering our utterances. Otherwise, it was not hard to do any of the entry steps.

Kore.ai gives you access to all the API data (and more) while you are testing in the interface. This is especially good because Kore.ai’s API also returns the most data, and you have access to data on individual words and analyses on sentence composition. Like Google, Kore.ai has a window-based system, so the supplemental windows for the chatbot can be moved around. The chat window itself is fixed on the right.

It was a little tough to nail down the authentication for Kore.ai’s API. It uses JWTs for authentication (essentially a payload of encrypted data), but it was difficult to identify what the contents of the JWT needed to be. We had to dig through the documentation to find and understand the correct syntax.
Qualitative Analysis of Each Platform continued

Business Considerations

Roadmap
Kore.ai provides a diverse set of features and functionality at its core, and appears to continually expand its offerings from an intent, entity, and dialog-building perspective.

Cost Structure
Kore.ai follows a session model. A usage session is defined as 15 minutes of user conversation with the bot or one alert session. The tier three plan carries an annual fee of $20,000, which includes up to 250,000 sessions. Enterprise licensing can be negotiated with Kore.ai.
Microsoft LUIS

Overall Platform Impressions
Microsoft LUIS has the most platform-specific jargon overload of all the services, which can cause some early challenges. The initial setup was a little confusing, as different resources need to be created to make a bot.

The look and feel is on par with AWS Lex, just with Azure styling instead. The color scheme is pleasing to the eyes, and important buttons are easily found and thoughtfully placed.

Advantages
Microsoft LUIS provides a simple and easy-to-use graphical interface for creating intents and entities. The tuning configurations available for intents and complex entity support are strong compared to others in the space.

Getting up and running in Microsoft LUIS is quick and easy. However, given the features available, some understanding is required of service-specific terminology and usage.
Qualitative Analysis of Each Platform continued

Downsides
At the core, Microsoft LUIS is the NLU engine to support virtual agent implementations. There is no dialog orchestration within the Microsoft LUIS interface, and separate development effort is required using the Bot Framework to create a full-fledged virtual agent.

Bot Framework Composer is an alternate option to custom development, as it provides a graphical drag-and-drop interface for designing the flow of the dialog. Microsoft LUIS provides an advanced set of NLU features, such as its entity sub-classifications. However, the level of effort needed to build the business rules and dialog orchestration within the Bot Framework should be considered.

Given that Microsoft LUIS is the NLU engine abstracted away from any dialog orchestration, there aren’t many integration points for the service. One notable integration is with Microsoft’s question/answer service, QnA Maker. Microsoft LUIS provides the ability to create a Dispatch model, which allows for scaling across various QnA Maker knowledge bases.

Setup, Training, and Testing
At the core, Microsoft LUIS is the NLU engine to support virtual agent implementations. The setup took some time, but this was mainly because our testers were not Azure users. Regular Azure users would likely find the process relatively straightforward. Once set up, Microsoft LUIS was the easiest service to set up and test a simple model.
Qualitative Analysis of Each Platform  continued

Entering training utterances was simple and straight forward. The entry flow was quick enough to keep up with our need to enter many utterances, which was helpful because the interface doesn’t provide a bulk utterance input option.

After publishing, Microsoft LUIS lets you compare your testing build with your published build for quick sanity checks and offers batch testing capabilities and intent tweaking right from the interface.

When testing intents in the API, publishing is a breeze. Once you train a model, you can publish it with two clicks. Endpoint URLs use GET parameters, so you can test them in your browser right away.
Qualitative Analysis of Each Platform continued

Business Considerations

Roadmap
Microsoft LUIS has displayed a decent pace of new features and enhancements to its existing functionality.

Cost Structure
Microsoft LUIS is priced on a per-transaction basis.

- For text requests, a transaction is an API call with a query length of up to 500 characters. The cost is $1.50 per 1,000 transactions.
- For speech requests, a transaction is an utterance with a query length of up to 15 seconds. The cost is $5.50 per 1,000 transactions.
NLU performance, your current platforms, and cost are all important factors to consider when choosing a conversational AI platform.
Summary

Each platform has strengths and weaknesses. This simple graph gives a summary of our score across each of the platforms:

**Evaluation Summary by Platform**

<table>
<thead>
<tr>
<th>Virtual Agent Vendor</th>
<th>Interface</th>
<th>Intents</th>
<th>Entities</th>
<th>Level of Effort</th>
<th>Roadmap</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Lex</td>
<td>★★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>Google Dialogflow CX</td>
<td>★★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>IBM Watson Assistant</td>
<td>★★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>Kore.ai</td>
<td>★★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>Microsoft LUIS</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★★</td>
<td>★★★★</td>
</tr>
</tbody>
</table>

Which platform is best for you depends on many factors, including other platforms you already use (such as Azure), your specific applications, and cost considerations. From a roadmap perspective, we felt that IBM, Google, and Kore.ai have the best stories, but AWS Lex and Microsoft LUIS are not far behind.

Need Help With Your AI Strategy?

Perficient has more than 30 dedicated AI delivery professionals ready to help you successfully execute an enterprise AI implementation. Our award-winning practice will help you turn your information into a strategic asset.

Our structured methodology helps enterprises define the right AI strategy to meet their goals and drive tangible business value. This four-phase approach addresses current state, business alignment, technology alignment, and developing a roadmap of candidate use cases.

Contact Us Today
Let Perficient help you on your digital transformation journey.

Perficient is the leading global digital consultancy helping transform the world’s largest enterprises and biggest brands. As a trusted end-to-end digital provider, Perficient partners with its Global 2000 and other large enterprise customers across North America to design and deliver digital transformation solutions that exceed customers’ expectations, outpace the competition and transform their business.

© PERFICIENT 2021