**Honda Virtual Assistant**

Bot Framework Analysis

**Members: Jason Xu, Jiankai Xu, Jeremy Lin**

**Table of Contents**

[Bot Framework Analysis 3](#_Toc55484490)

[Rating Criteria 3](#_Toc55484491)

[Quality Analysis 3](#_Toc55484492)

[Knowledge 3](#_Toc55484493)

[SLA (Service Level Agreement) 5](#_Toc55484494)

[Development Analysis 6](#_Toc55484495)

[Development Cost 6](#_Toc55484496)

[Maintenance Cost 8](#_Toc55484497)

[Complexity 10](#_Toc55484498)

[Adaptability Analysis 12](#_Toc55484499)

[Requirements 12](#_Toc55484500)

[Environment Analysis 13](#_Toc55484501)

[Support Teams/Skype 13](#_Toc55484502)

[Language Support 13](#_Toc55484503)

[Decision Making 15](#_Toc55484504)

[Analysis Chart 15](#_Toc55484505)

[Licensing Cost 16](#_Toc55484506)

[Conclusion 17](#_Toc55484507)

[Comparison 17](#_Toc55484508)

[Demonstration Decision 17](#_Toc55484509)

[Final Decision 18](#_Toc55484510)

[References 19](#_Toc55484511)

# Bot Framework Analysis

## Rating Criteria

*Table 1: Table of explanation of analysis rating*

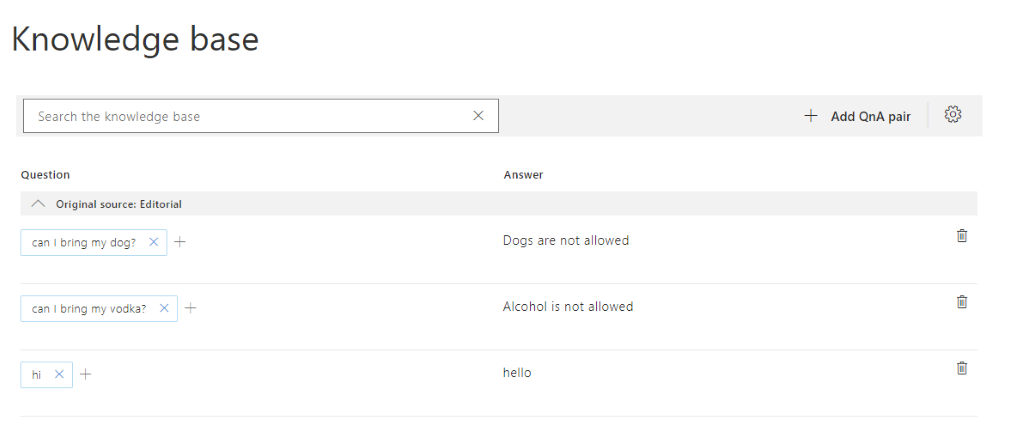
|  |  |
| --- | --- |
| **Rating** | **Description** |
| 1 | Bot framework cannot achieve this factor/requirement. |
| 2 | Bot framework performs poorly on this factor. |
| 3 | Bot framework can achieve basic requirement, but not complete. |
| 4 | Bot framework works excellent on the factor with minor drawback. |
| 5 | Bot framework acts perfectly on this factor. |

## Quality Analysis

### Knowledge

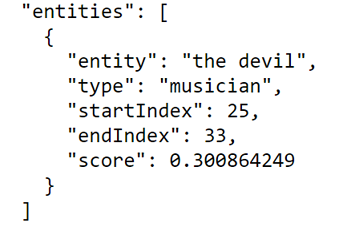
* Azure (Rating: 5)

The response quality of Microsoft Azure is assured if you have a bunch of knowledge base documents to feed the bot. The knowledge base connection is through qnamaker.ai, which is an integral part of Microsoft cognitive science services.



*Figure 1: QnA Maker Pair Layout*

If you integrate with LUIS, which is language understanding service provided by Microsoft, you have the chance to develop a bot that can based on user’s input message, dispatch the message to either QnA maker for answer searching, or to the LUIS, for capturing user’s intent and direct the user a way of solving the problem.



*Figure 2: LUIS entities*

For example: If the user says: how to submit a ticket? The bot will analyze the user’s intent as to get an answer (QnA), and then search in the knowledge base for the best match answer. If the user says: submit a ticket. The bot will analyze the user’s intent as specific entities with score. Then, based on design, provide ways for the user to submit a ticket. In this case, the bot might reply: “OK, here you can file a ticket”, “Tell me about your issue.”. The user will then be able to interact with the bot to complete the ticket filing.

* Amazon Lex (Rating: 5)

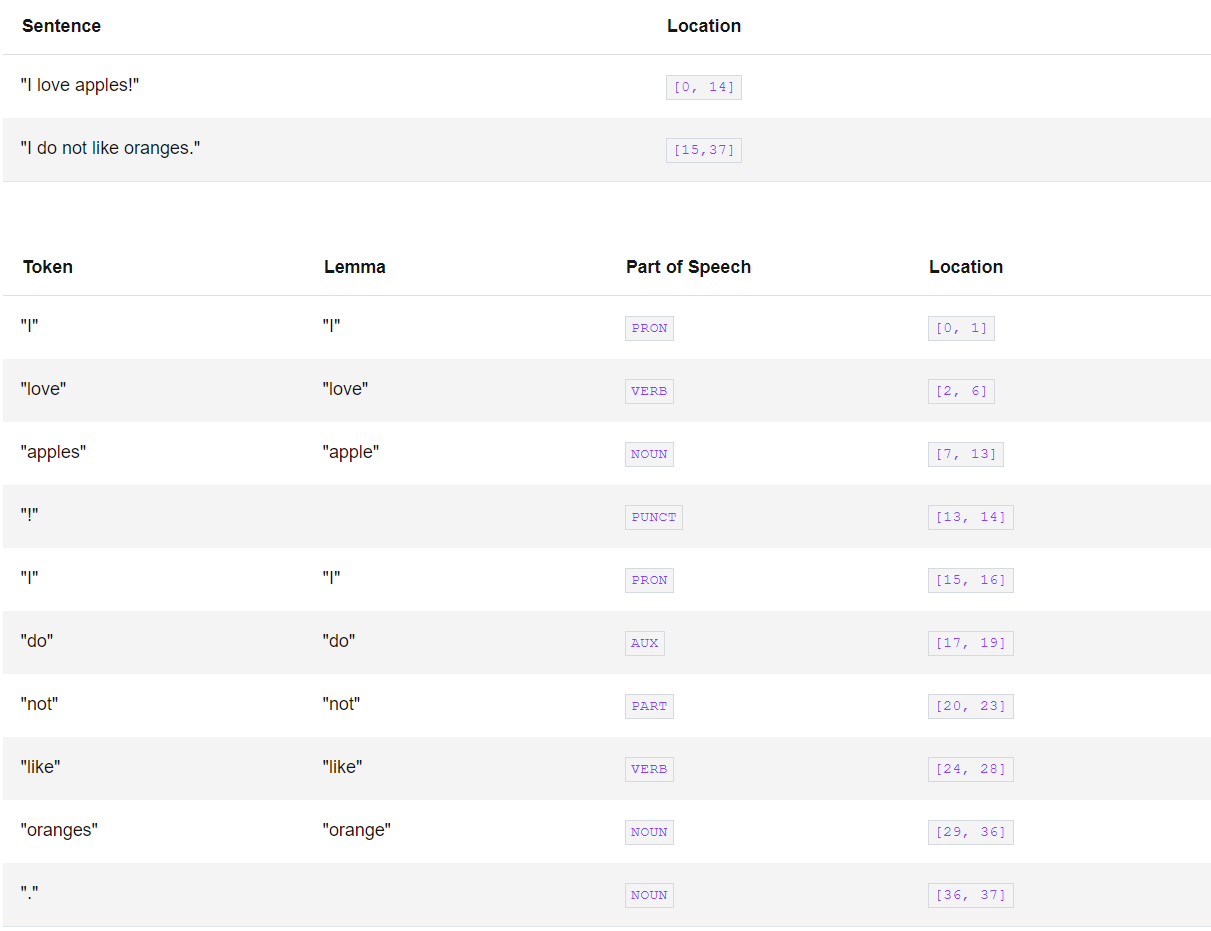
AWS has strong language understanding techniques. The framework comes with built-in natural language understanding (NLU) capabilities and machine learning from AWS. These features help your bot to accurately interpret user intent and to become smarter and more capable as its ages.

* Pandorabot (Rating: 5)

Pandorabot primarily uses AIML to configure the bot’s knowledge base. Intents, topics, synonyms can all be configured using XML-like mark up.

* IBM Watson (Rating: 5)

With IBM Watson™ Natural Language Understanding, developers can analyze semantic features of text input, including categories, concepts, emotion, entities, keywords, metadata, relations, semantic roles, and sentiment. (IBM, 2020)



*Figure 3: IBM Natural Language Understanding Syntax*

* Botpress (Rating: 5)

Its NLU module includes: Intent recognition: recognizing what the user wants; Entity extraction: extracting structured information from messages like dates, time, cities, names and more; Slot tagging: identify necessary parameters to fulfill given task; Language identification: knowing in which language the user is writing. (Botpress, 2020)

* Google Dialogflow (Rating: 5)

Google Dialogflow is equipped by a strong machine learning backend which contains NLP capabilities for teasing user intents out of any given utterance. Machine learning that trains your bot to deliver increasingly valuable responses (built on its engagement with users).

### SLA (Service Level Agreement)

* Azure (Rating: 5)

The SLA guaranteed 99.9% of the time will successfully receive and respond to REST API calls to the Premium Channels API Endpoint. (Microsoft, 2017) Guaranteed 99.9 % or greater availability.

* Amazon Lex (Rating: 5)

Amazon do not have a specific service level agreement for the Lex product. However, Lex is a part of amazon AWS, one of the world’s largest cloud computing service providers. (Amazon, unknown)

* Pandorabot (Rating: 5)

99.9%+ uptime guarantee. Phone, email, and public office hours for enterprise edition (recommended if there will be more than 100,000 interactions per month).

* IBM Watson (Rating: 5)

IBM Watson guarantees 99.99%+ uptime.

* Botpress (Rating: 5)

24/7 support available for your global business with SLA. Plus, it provides online or in-person training

* Google Dialogflow (Rating: 5)

Google also claims that Dialogflow maintains a >=99.9% of monthly uptime percentage. (Google, 2020)

## Development Analysis

### Development Cost

* Azure (Rating: 4)

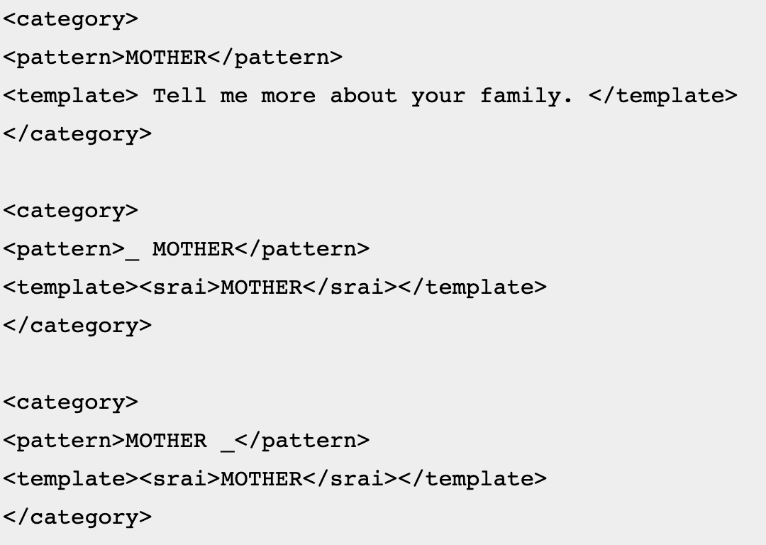
To set up a basic Azure bot with QnA maker is simple. With the strong knowledge base documents backend, the bot should perform well in the question answering role. However, Luis requires a cumbersome manual training, annotating, revising process if there are no existing models for the intents. You might need to feed up to thousands of messages to train for a single intent and annotate tens or hundreds of nouns to train an entity classifier. Whatsoever, almost every bot framework needs that so not a minus.

* Amazon Lex (Rating: 4)

The advantage of Lex is that it provides Full-featured NLU and machine learning capabilities, SDKs in multiple languages, and a strong backend service supporter, AWS. However, compared to Microsoft Azure, Lex is not an important/core product of AWS, it’s more like a side product of Alexa. The documentation is provided but not that clear and thorough. Developers need to first get used to the AWS development platform, and then go to the bot itself because many functionalities are strongly connected to the AWS platform.

* Pandorabot (Rating: 4)

Unlimited sandbox messages: Pandorabot primarily uses a markup language called AIML (an XML dialect) to configure topics, responses, synonyms. The official SDKs support Java, Python, Node.js. Additional community developed open-source SDKs also include support for Ruby, PHP, and Golang. Depending on personnel familiarity with those tools may or may not increase development cost.



*Figure 4: AIML Example*

* IBM Watson (Rating: 4)

It comes preloaded with pre-configured language models (e.g. customer care, banking) that could give you a head start building your bot. Supported languages for the IBM Watson SDK include Python, Java, GoLang, Node.js, Swift. However, without direct integrations to Microsoft Teams, there would be an increased development cost.

* Botpress (Rating: 4)

Since the platform provides their own executable, and the manage desk is displayed on local server, therefore no other required knowledge on other software is needed. It also provided open-source version, as well as the UI only version. A plus is that the Botpress also provides debug package.

Botpress is also famous for its user-friendly, graphical interface, flexibility, and customization. But the learning curve can be high, this tool is not for part-timers or those who want to build basic, scripted chatbots. It is ideal for start-ups and agencies with limited resources that need to build great bots. And it is hard to run various instances from one installation.

* Google Dialogflow (Rating: 4)

Dialogflow is easy to start on. Unlike Amazon Lex that overly rely on other AWS services to perform functionalities, Dialogflow does not require too many other google services to enable the bot to perform well. Although it is easy to start, to build a sophisticated agent requires much work of building up intents, entities, and contexts.

### Maintenance Cost

* Azure (Rating: 5)

Flexible support plans starting at $29/month. Other than pricing, there are multiple ways to debug the bot: using IDE, developer can debug the source code by setting break point; developer can also debug with Bot Framework Emulator. The Emulator provides the JSON information includes key metadata, and the system log of services trace.

The LUIS model provides the possibilities of feeding more training data to the system.

Graphical user interface, text, application

Description automatically generated

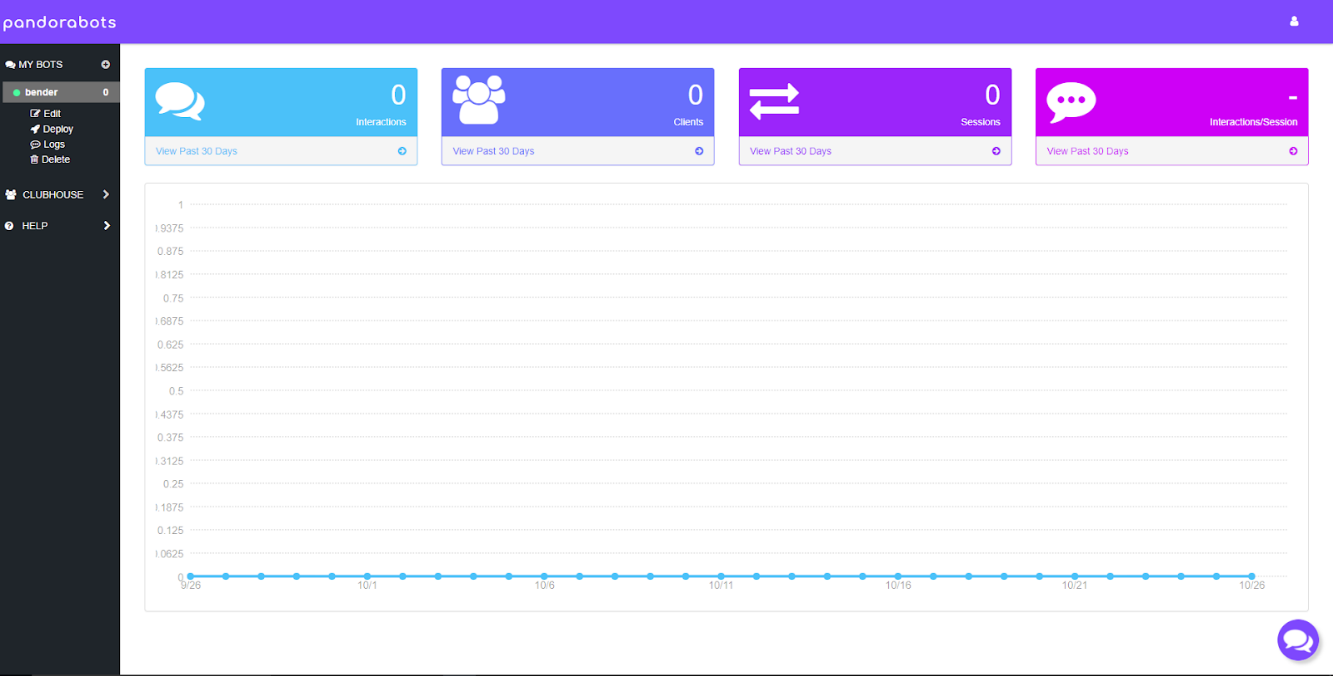
*Figure 5: Azure Bot Framework Emulator*

* Amazon Lex (Rating: 5)

AWS provides developer, business, and enterprise support plan, which has $29/month, $100/month, and $15000. The plans cover different kinds of services that cover many types of problems.

* Pandorabot (Rating: 5)

Supports CI/CD and staging environments. Allowing developers and maintainers to test code in a production-like environment before users have a chance to use updates. This could help reduce maintenance costs. Pandorabot also provides a dashboard to obtain analytics on bot interactions. Pandorabot also saves chat logs for anonymized users.



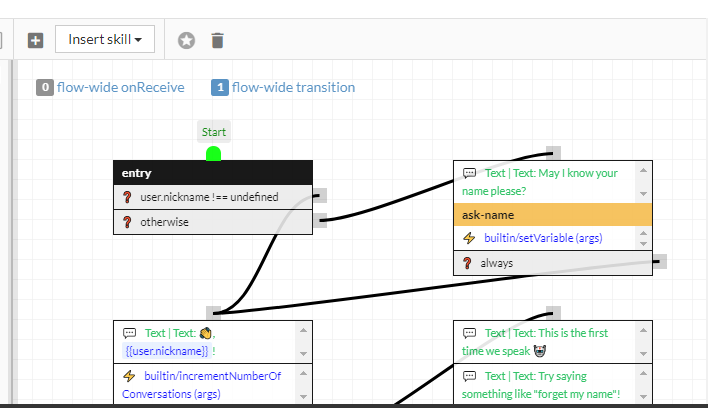
*Figure 5: Pandorabot Analytics Dashboard*

* IBM Watson (Rating: 4)

IBM Watson comes with an activity tracker that makes it easy to investigate incidents and related bot statistics. Unfortunately, IBM Watson texts do not get mapped to intents in provided metrics which is a limitation to maintenance efforts.

* Botpress (Rating: 5)

After early-stage development, the non-developer can edit the flow chart via UI version, which is convenient and straightforward. However, the non-developer still needs to understand the structure. But it provides 24/7 support and online in-person training. That may even it out.



*Figure 6: Botpress Flow Editor*

* Google Dialogflow (Rating: 5)

Support plans starting from $100-$250 per month. Google has a strong support team which could assist in the development and maintenance.

### Complexity

* Azure (Rating: 4)

Might be a minus for Azure. The whole framework can be implemented in two ways: in SDK, you code your bots; and in composer, you build your bot using pure UI without coding. We have been using SDK and find that you must have some basic coding experience to take on. Although the documentation provides code samples, you must take some time to think about how it works. Also, if you want to build something outside of functionalities covered by the documentation, it would be difficult. One important part of Azure bot is that its JavaScript version is hard to debug. (Using SDK) Using composer, on the other hand, would be free from coding bugs, but might encounter less satisfied performance training from scratch. (Common problem for all bot frameworks using UI)

* Amazon Lex (Rating: 3)

Lex is somewhat inclined to speech recognition (because of Alexa, an amazon core product) If text interaction is enough, Lex cannot show its advantage. For language understanding, the same process as other bot frameworks (add intent, entity...) is required.

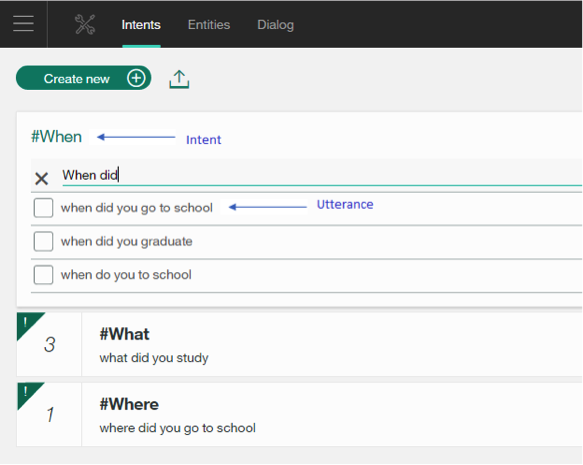
If you want to create a custom bot, you must first create and train a lambda function which directs the flow of the bot at Amazon Lambda, and then create an intent, create slot types, and configure the intent. Since it is highly integrated with AWS and its affiliated services, it is more complex than other frameworks that only provide chatbot services.

* Pandorabot (Rating: 4)

Pandorabot makes it easy to deploy to web. Web UI provided, so even CX designers can get involved with engineering of bots. However, the web UI, again, requires understanding of markdown language which may add an increased learning curve.

* IBM Watson (Rating: 5)

IBM Watson provides intent defining interfaces that are made usable to non-engineers. Furthermore, intents can be used in conjunction with webhooks for more robust integrations.



*Figure 7: IBM Watson Intent Defining Interface*

* Botpress (Rating: 4)

Botpress has been built for developers and, it is not for everyone. You need to have some skills in programming and be familiar with open-source projects (both skills can be easily learned). Botpress is written in JavaScript, but you do not need to be an expert. With basic skills in programming, you will be fine.

Like most other bots, the Botpress provide open-source code on GitHub, as well as their own executable and UI management desk. However, for language part, it only supports JS. And even though it is easier for non-developer to maintain the system after the development, it is still more developers oriented.

* Google Dialogflow (Rating: 5)

Dialogflow has a clear documentation with video tutorials for different concepts the user needs to know to use it. Google also provides substantial number of prebuilt agents which the user can start with. Building a Dialogflow bot does not require you to have any coding experience and machine learning knowledge. All operation in the building stage is UI-based which is direct and clear.

## Adaptability Analysis

### Requirements

* Azure (Rating: 5)

Microsoft Azure bot framework can be deployed to many channels. In Honda’s case, Azure bots can be added to Microsoft Teams and Skype. The deployment process is a little bit complex, but it is doable.

* Amazon Lex (Rating: 4)

Amazon Lex can be deployed to several message platform, web application, and mobile. Connected to other AWS services, it could also perform tasks like call center agent assistant. A response card can also be involved in the conversation. Benefiting from a various service provided by AWS, Lex can achieve more than just a single chatbot. The problem is that it overly relies on other AWS services, and you must adopt AWS services if you want Lex to do more.

* Pandorabot (Rating: 2)

Pandorabot is not very adaptable for our functional requirements. Pandorabot lacks direct integrations with Skype and Microsoft Teams. However, pandorabot does provide context understanding capabilities for more complex conversations.

* IBM Watson (Rating: 5)

IBM Watson comes with data isolation for our security functional requirements. Activity tracking for maintenance and debugging. It also comes with skill customizations and learning capabilities.

* Botpress (Rating: 4)

It relies on the host via other server, like the Azure. However, it can be deployed on Azure side, and the rest of deployment work like azure bot. However, other than the chat bot side, it may be hard to implement other requirements like analysis, card function with Teams interface. From this point, Botpress serves only as a chatbot.

* Google Dialogflow (Rating: 4)

Dialogflow can provide rich response messages such as cards, and you can even customize your response. Since the Google Dialogflow is actually deployed on the Google Cloud Platform, it could provide usage analysis and error reporting, and further integration with other google products easily. However, Dialogflow provides no direct integration with Microsoft Teams. It does provide integration with Skype, but that is to be deprecated soon.

## Environment Analysis

### Support Teams/Skype

* Azure (Rating: 5)

Azure bot framework can directly deploy on both platforms, which is a big plus. Although the deployment process might encounter some account issues, the direct official support is much more convenient than all its opponents.

* Amazon Lex (Rating: 3)

Although Lex can be integrated with AWS services seamlessly, it needs a vehicle to connect to Microsoft Teams or Skype. In this case, Azure portal is used to connect the Lex bot to Microsoft Teams. (Zhang W., 2020)

Unless your corporation is using AWS family software and services, Lex is not highly recommended.

* Pandorabot (Rating: 2)

No direct integration with Microsoft Teams and Skype. Pandorabot however can be deployed on other cloud platforms such as Heroku.

* IBM Watson (Rating: 4)

Difficulties in integration with other ecosystems (Google, Microsoft, and Amazon) i.e., not the best option when targeting skype or Microsoft Teams client environment. However, tools exist to bridge this gap (see [tray.io](https://tray.io/connectors/ibm-watson-microsoft-teams-integrations)).

* Botpress (Rating: 4)

It relies on the host via other server, like the Azure. However, it can be deployed on Azure side, and the rest of deployment work like azure bot. Other than the chat bot side, it may be hard to implement other requirements like analysis, card function with Teams interface. From this point, Botpress serves only as a chatbot.

* Google Dialogflow (Rating: 4)

It could be deployed to Skype which is used by Honda. However, the deployment requires the involvement of Azure. Dialogflow requires you to create a bot in azure, and set the message endpoint to Dialogflow, and then deploy the bot in Azure’s channel. Its performance might be affected by your Azure’s subscription if you have a high volume of requests. In this part, Azure is a plus. Also, Azure is no longer supporting large scale deployment on Skype, such as adding to a group more than 100 people.

### Language Support

* Azure (Rating: 5)

Multilingual, and support more than 15 languages in its language understanding services. Japanese is also supported.

* Amazon Lex (Rating: 1)

Lex only supports English and Spanish. In our case, languages like Japanese is not supported, so Lex is not something we are looking for.

* Pandorabot (Rating: 4)

Pandorabot allows creating for multilanguage support in AIML scripts. It may be tedious to create individual AIML scripts for different languages.

* IBM Watson (Rating: 5)

IBM Watson has multilanguage support, which may be useful given that Honda is a multinational organization. Included languages are English, Japanese, Spanish, German, and more.

* Botpress (Rating: 5)

Botpress supports 12 languages, including Japanese.

* Google Dialogflow (Rating: 5)

Support more than 15 languages including Japanese. A multilingual agent is also supported with one root language, and a local-specific language.

# Decision Making

## Analysis Chart

Table 2 provides our rating on 6 market mainstream bot framework, including Azure bot framework, Amazon Lex, Pandorabot, IBM Watson, Botpress and Google Dialogflow. This analysis table is inspired by Amol Govindwar (Govindwar A., 2020) from Honda. Some modification has been made to fit our analysis better.

The first factor to consider is quality, which includes the knowledge of the bot framework. This means the NLP (natural language processing) level, and other related attributes of language processing. The other factor is SLA. It is determined by the level of support that can be provided by the bot framework in the contract.

In the development factor, three options are discussed. The first one is development cost, which involves the effort and difficulty of developing the bot framework. The second one is maintenance, which represents the hardness of maintaining the bot framework after development. The last one is complexity. Complexity embodies the developer’s view on the bot framework. This factor overlaps with the development cost to some degree, but some area like UI layout and friendliness throughout the development has not been mentioned in the development cost. In other words, the development cost may orient more on the architecture side of the system.

The adaptability factor, which is the most significant part, determines whether our bot framework can acclimate with the prioritized requirements, which include security requirement and incident requirement, of our system.

The last factor is considering about whether the bot framework can suit our environment requirements, that is whether the bot framework can integrate with Microsoft Teams channel and Skype channel, as well as whether the system can understand multiple languages other than English, in that Honda is an international company. Specific language support should be considered is Japanese.

The rating inside the table 1 is ranged from 1 to 5. The rating criteria is provided on the first chapter. For all factors, the higher the rating is, the better impact on bot framework to fit the requirement is. To clarify, in the ‘development cost’ field, Azure has a rating of 4.5 while Amazon Lex has a rating of 4. This means that Azure has less development cost than Amazon Lex, because in this case, higher rating means better performance. The explanation of rating is described in table 2.

Comparing to Amol Govindwar’s original analysis chart (Govindwar A., 2020), our decision is to take the license cost out of the chart. Since we cannot dig into all 6-bot framework, as well as the pricing of all framework depends on the level of customization, it is hard for us to determine the specific rating on this field. Enterprise pricing often varies and requires a call to the interested provider for a quote. Since the rating cannot be accurate, we decide to discuss this separately.

*Table 2: Table of bot framework analysis*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Factors to Consider** | | **Bot Framework** | | | | | |
| Azure | Amazon Lex | Pandorabot | IBM Watson | Botpress | Google Dialogflow |
| **Quality** | Knowledge | 5 | 5 | 5 | 5 | 5 | 5 |
| SLA | 5 | 5 | 5 | 5 | 5 | 5 |
| **Develop-ment** | Development Cost | 4 | 4 | 4 | 4 | 4 | 4 |
| Maintenance Cost | 5 | 5 | 5 | 4 | 5 | 5 |
| Complexity | 4 | 3 | 4 | 5 | 4 | 5 |
| **Adaptabi-lity** | Requirements | 5 | 4 | 2 | 5 | 4 | 4 |
| **Environ-ment** | Support Teams/Skype | 5 | 4 | 2 | 4 | 4 | 4 |
| Language Support | 5 | 1 | 4 | 5 | 5 | 5 |
| **Result** | | 38 | 31 | 31 | 37 | 36 | 37 |

\*Ratings range from 1-5. Higher rating means greater contribution to that factor.

## Licensing Cost

*Table 3: Table of Bot Framework Pricing*

|  |  |
| --- | --- |
| **Bot** | **Pricing** |
| Azure | Premium: Free for 10,000 message/month or $0.50/1000 messages.  Standard Channels: Unlimited messaging for free. Pay for what is used.  The QnA maker is free, the LUIS is $1.50 per 1,000 transactions. |
| Amazon Lex | $0.00075 per text request. Pay for what you use |
| Pandorabot | For enterprise level pricing determination, we would have to contact Pandorabot.  $199/month for 100,000/month channel messages and $2.50/1,000 Messages for additional channel messages |
| IBM Watson | License cost is to vary by enterprise. |
| Botpress | Community – Free; Pro - from $45/month; Custom - Custom fee |
| Google Dialogflow | ES: $0.002 per request, free for knowledge base connectors.  CX: $20 per 100 chat sessions (every 40 request or 30 mins after last request) |

Among 6 bot framework we have picked, only 3 of them (Azure, Amazon Lex, and Google Dialogflow) have specific pricing details listed. The rest of them provide little information about the pricing because the pricing is based on the level of customization. In this case, it is hard for us to compare all the licensing cost without delving into all 6-bot framework we have picked. With this short developing period, we find it hard to achieve this goal, therefore we decided to discuss about the licensing cost based on the information we had, including official data and internet reviews.

## Conclusion

### Comparison

Comparing the rating on the factors, we note that the quality level on mainstream bot framework is quite high. Note that all bot frameworks have developed complete knowledge system, including language processing and human-like interaction. Considering SLA, all companies provided exhaustive document, well-established customer support and consultant based on the customization requirements.

For development factors, all bot frameworks have ratings of 4. Even they provided UI interface for user to build the system better, some basic skills are required to understand the framework. Frameworks are also written in specific language, for example, Azure supports C# and JavaScript, while Botpress is written in JavaScript only. This may add up the programming language requirements. For maintenance cost, all bot frameworks provide UI layout of system, to help the post-development maintenance. Botpress provides flow chart to help CX designer to add up language logic, while Pandorabot has the same thing. Talking about the complexity, we consider this factor more on developer side, the reason of this is that for bot framework, it does not have much complexity considering the user side. Less challenge is presented with the bot framework because except Amazon Lex, it provides function of chatbot only. Amazon Lex involves AWS services, which increase its degree of complexity.

In the adaptability and environment support, we would like to adapt the bot framework with other requirements we have discussed in the workbook, including login validation, routine task identifier and incident report. Note that expect for Pandorabot, all the other frameworks have a direct way or bridge tool to connect with either Microsoft Teams or Skype. However, the reason that only Azure gets a five, it is because that bridge tools can be buggy sometimes, and it decreases the integrality of the system. Note that only Azure, Google Dialogflow, and IBM provides the host for the chatbot, the other framework needs those host services to deploy their bots. For example, Botpress needs to register for a host to be deployed on Teams. This is a minus of the performance on adaptability. The environment factors of supporting existing Microsoft Teams and Skype has the same reasoning with adaptability. For language support, since Honda is a Japanese company, we took Japanese into our consideration. Except for Amazon Lex, all other frameworks support Japanese. However, for Pandorabot, based on its AIML structure, it may be hard to translate the tags into Japanese.

### Demonstration Decision

Based on the analysis in part I, our decision making is based on the rating summation of given factors. Since Azure has the highest rating and the IBM Watson and Google Dialogflow have the second, on the inception and elaboration stage of our development, we decided to build demos based on Azure and Google Dialogflow based on the fact that IBM Waston and Azure have a lot in common and our goal is to explore the differences between those mainstream bot frameworks.

### Final Decision

We decided to develop on Azure based on:

1. Azure can be deployed on both Microsoft Teams and Skype, while Google Dialogflow can be only deployed on Skype. Note that the Microsoft decided to commit their focus on Microsoft Teams, focusing on the framework that is compatible with Teams will be wise in the long run.
2. Azure can be modified and customized better by editing the source code of the bot, while Dialogflow only provides UI interface to the user. Such freedom is a plus on the development phase.
3. Azure acts better on the Microsoft Teams app function. For example, within the chat, Azure’s response can implement Teams card API. With this function, we can cooperate our requirements with the framework better.
4. Honda is using Azure services for the moment. It may be easier to integrate the system better in the future. Plus, experiences on the service can be shared to improve the development.

# References

Govindwar A. (2020, Sep 17). *Criteria for Chatbot*. Honda.

Botpress (2020, Oct 27). *Developer’s Guide 12.10.10*. Botpress Inc.

Microsoft (2017, Dec). *SLA for Azure Bot Service*. Microsoft. URL: <https://azure.microsoft.com/en-us/support/legal/sla/bot-service/v1_0/>

Amazon (Unknown). *AWS Service Level Agreements (SLAs)*. Amazon. URL: https://aws.amazon.com/legal/service-level-agreements/

Google (2020, Sep 10). *Dialogflow Service Level Agreement (SLA)*. Google. URL: <https://cloud.google.com/dialogflow/sla>

Zhang W. (2020, Mar 6). *A cleverer chatbot with Amazon Lex in Microsoft Teams*. LinkedIn. URL: <https://www.linkedin.com/pulse/cleverer-chatbot-amazon-lex-microsoft-teams-wuming-zhang>

IBM (Unknown). *Pricing for IBM Watson Assistant*. IBM. URL: <https://www.ibm.com/cloud/watson-assistant/pricing>

IBM (2020). *Natural Language Understanding*. IBM Cloud Docs. URL: https://cloud.ibm.com/docs/natural-language-understanding?topic=natural-language-understanding-about