

# Development and Research of a Chatbot Using the Linguistic Core of Amazon Lex V2

Victoriia Hnatushenko, Kateryna Ostrovska and Valerii Nosov

*Ukrainian State University of Science and Technologies, 4 Gagarina Ave., Dnipro, 49600, Ukraine*

## Abstract

The main of this research is to develop and explore the configuration of a text and voice recognition system, integrate it into a specialized application, and deploy the application in a cloud environment. Amazon Lex service is built on chatbots that support Natural Language Understanding (NLU) and voice recognition. The developed chatbot elevates the user experience while engaging with voice consultants by offering flexible customization options. A chatbot has been designed with interactive text input fields and voice recording functions. The server architecture of the application is configured for seamless data transmission through the AWS SDK to Amazon Lex. The input information undergoes processing to ensure the generation of responses that are dynamically displayed on the web page. The structure of all intents – simulating banking services such as checking card balance, transaction history, and more. Testing the intents was done by creating a dataset with possible user statements and automated runs. The developed chatbot was tested through 6 runs, each consisting of up to 5 statements for recognition. The accuracy of text input recognition ranged from 60% to 99%, with voice input recognition accuracy being 10% lower.

## Keywords

Amazon Lex V2, Amazon Web Services, natural language, artificial intelligence, recognition, chatbot

## 1. Introduction

In recent times, there has been a surge in interest surrounding the integration of voice and text assistants across various sectors, particularly in business [1, 2] and media. This evolving technology holds immense relevance and is advancing at a rapid pace. However, the landscape for creating one's own chatbot is rich with diverse solutions. Among the plethora of Artificial Intelligence (AI) services available, Amazon Lex V2 stands out for its exceptional natural language recognition capabilities.

Built upon chatbots that excel in Natural Language Understanding (NLU) and voice recognition, Amazon Lex V2 offers a comprehensive suite of features. Being a part of the Amazon Web Services (AWS) ecosystem, it seamlessly integrates with other services within the platform, facilitating effortless deployment directly onto the cloud and automatic incorporation into serverless architectures.

Chatbots are a class of intelligent, conversational software algorithms activated by natural language input. They can intelligently respond to inputs, understand commands and execute tasks [3].

Researching Lex, developing and integrating a customized bot provides the opportunity to understand the general logic behind the functioning of similar services. It allows for analyzing the quality of text and voice recognition, as well as assessing the practicality of its application across various types of software solutions, such as a banking assistant. This exploration can offer valuable insights into the efficiency and effectiveness of integrating such technology into different contexts, enhancing user experience and optimizing task performance.

---

COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems, April 12–13, 2024, Lviv, Ukraine

✉ vvitagnat@gmail.com (V. Hnatushenko); kuostrovskaya@gmail.com (K. Ostrovska); valex.nosov@gmail.com (V. Nosov)

🆔 0000-0001-5304-4144 (V. Hnatushenko); 0000-0002-9375-4121 (K. Ostrovska); 0009-0003-5841-8995 (V. Nosov)



© 2024 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Key features of Amazon Lex V2 include [4]:

1. **Natural Language Understanding (NLU):** Amazon Lex V2 uses sophisticated algorithms to understand and interpret natural language input from users. This allows for the creation of conversational experiences that feel intuitive and human-like;
2. **Voice Recognition:** The service supports voice recognition, allowing users to interact with applications using spoken commands. This feature enables hands-free interaction and accessibility for users with disabilities;
3. **Multi-turn Conversations:** Amazon Lex V2 supports multi-turn conversations, where the bot can engage in a back-and-forth dialogue with users to gather information or fulfill requests. This capability enables more complex and interactive interactions;
4. **Integration with AWS Services:** As part of the AWS ecosystem, Amazon Lex V2 seamlessly integrates with other AWS services, such as Lambda functions for backend processing, DynamoDB for data storage, and Amazon Connect for contact center solutions;
5. **Customization and Scalability:** Developers can customize the behavior and responses of their chatbots using Amazon Lex V2's flexible configuration options. Additionally, the service is designed to scale automatically to handle varying levels of traffic and usage.

## 2. Related works

The documentation for Amazon Lex V2 provides a comprehensive amount of information regarding the bot's functionality and integration logic. However, its effectiveness compared to other AI services raises the most questions. Among recent publications, an article on the American portal Medium [5] introduces the concept of conversational AI matrices, outlining a general rating of commercial natural language recognition systems (Figure 1).

Chatbot Offerings Rating Matrix	NLU		Graphic Dialog Node Management	Native Code Dialog Node Management	Machine Learning	Hosting	Cost	NLU API Capability	Enterprise Ready Scaling
	Intents	Entities							
IBM Watson Assistant	↗	↗	↑	↗	↗	Commercial Cloud	→	↗	↑
Amazon Lex	↗	↗	↓	↑	↗	Commercial Cloud	↗	↗	↗
Microsoft Azure LUIS Bot Framework Composer / Emulator	↗	↑	↗	↑	↗	Commercial Cloud	↗	↗	↑
Microsoft Power Virtual Agent	→	→	↗	↗	→	Commercial Cloud	↓	↓	→
Rasa	↑	↗	↑	ML Approach	↑	Install Anywhere	Open Source	↑	↑
Cisco MindMeld	↗	↗	↓	↗	↗	Install Anywhere	Open Source	↗	→
Google DialogFlow ES	↗	↗	→	↑	↗	Commercial Cloud	↗	↗	↗
Google DialogFlow CX	↗	↗	↑	↓	↗	Commercial Cloud	↓	↗	↗
NVIDIA Jarvis	↑	↑	↓	↓	↑	Install Anywhere	Dedicated Hardware	↗	↑

↑ Exceptional   ↗ Good   → Average   ↓ Poor

**Figure 1:** Conversational AI Rating Matrix

While Amazon Lex undeniably excels in natural language understanding, custom code execution, and machine learning capabilities, its integration with the Amazon Web Services (AWS) cloud infrastructure presents certain limitations. Although AWS offers an array of convenient tools for development and seamless integration with other services, this reliance on a specific cloud environment could be perceived as restrictive, as highlighted by the author of the article. Nonetheless, many proponents argue that the benefits of leveraging the AWS ecosystem often outweigh these constraints.

One significant drawback highlighted in the article is the perceived limited functionality of Lex's visual chatbot constructor. However, it's essential to acknowledge that this assertion may not fully reflect the reality. While compared to Google DialogFlow, Lex's constructor may have a narrower scope of features, it nonetheless exists and is continuously evolving, particularly in its V2 iteration [6].

In conclusion, the article emphasizes the specific use cases where Lex V2 may excel, such as in call centers or as a virtual consultant. However, it's crucial to recognize that Lex's capabilities extend beyond these applications, and its integration within the AWS ecosystem offers unique advantages for developers seeking robust conversational AI solutions. As Lex V2 continues to evolve, it is likely to address and overcome many of the current limitations, further solidifying its position as a leading platform in the field of conversational interfaces [6].

In another publication on the developers' platform dev.to [7], the limitations of Amazon's Lex linguistic core are brought to light in comparison to advanced technologies like ChatGPT and other implementations featuring OpenAI integration. The article underscores Lex's primary challenge as the constraints in utilizing intents, noting that at least 100 utterances are required for each intent to encompass all potential user questions effectively. Furthermore, the author points out that even minor word permutations can significantly influence the determination of the intent used, contributing to a notable dilemma. Hence, attaining high-quality recognition necessitates exhaustive exploration of various question-answer permutations to comprehensively cover all possible communication pathways.

While Lex remains a robust platform for constructing conversational interfaces, it evidently exhibits limitations in terms of intent recognition when juxtaposed with cutting-edge language models employing OpenAI's advanced technology. This sheds light on the continuous evolution in natural language understanding and the burgeoning demand for sophisticated, context-aware conversational AI systems [7].

This dev.to article underscores the critical importance of advancing linguistic cores and intent recognition within conversational AI, driving innovations aimed at addressing the complexities of natural language understanding and meeting the escalating expectations for nuanced and adaptive conversational interfaces [7].

An interesting application of Amazon Lex is revealed in the publication on the Toolify.ai website [8]. The author's project aims to develop a chatbot using Amazon Lex, which can effectively communicate with users and gather information about their physical symptoms. By utilizing machine learning algorithms and predictive models, the chatbot can analyze symptoms provided by users and offer predictions regarding potential illnesses they may have.

The chatbot operates by interacting with users using predefined prompts and questions. Initially, it inquires about users' most severe symptoms and then asks about any other mild symptoms they may be experiencing. The chatbot records and stores users' responses in slots, which are special fields used for gathering information [8].

In conclusion, the use of Amazon Lex in developing a chatbot for effectively communicating with users and gathering information about their physical symptoms demonstrates the potential for leveraging machine learning and predictive models in healthcare applications. The ability to analyze user-provided symptoms and offer predictions regarding potential illnesses showcases the practical implications of such technology in assisting and informing individuals about their health. This approach not only illustrates the advancements in artificial intelligence but also highlights the potential for improving healthcare interactions through chatbot technology [8].

It is important to note that the issue of using NLP and bots in medicine is being discussed in the Jamda journal, which explores the use of natural language processing in the post-treatment period [9].

### **3. Methods and materials**

The purpose of the research is to examine the Amazon Lex V2 chatbot and test its natural language recognition mechanism. Based on the test results, an analysis is conducted to determine

the advantages and disadvantages of this platform as well as the feasibility of its application in the banking context. The study focuses on Lex V2 and the cloud environment of Amazon Web Services with the aim of evaluating the performance and usability of the chatbot in the banking industry. It should be noted, chatbots are able to provide exactly this: a more convenient, interactive and unique alternative to traditional customer service [10]. Configuring the bot requires preparing all necessary infrastructure and setting up a static web page, ensuring a robust and reliable setup for optimal user interaction in the banking domain.

The plan is to develop a banking chatbot to simulate customer interactions with banking support services. Its main aim is to provide intuitive assistance for various banking inquiries using advanced AI technology.

Lex uses a combination of machine learning and natural language processing algorithms to understand and process user input. The platform leverages advanced algorithms for intent recognition, entity recognition, and context management within conversational interactions. While the specifics of the underlying algorithms are proprietary to Amazon, it's known that Lex incorporates deep learning techniques and models such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and other state-of-the-art NLP methodologies to interpret and respond to user inputs effectively. Additionally, Lex may utilize technologies such as word embeddings, attention mechanisms, and sequence-to-sequence models to enhance its language understanding capabilities. These techniques enable Lex v2 to provide accurate intent classification and entity extraction, contributing to its robust conversational abilities.

RNNs have found wide-ranging applications in speech and language processing due to their ability to handle sequential inputs of varying lengths, a task that traditional feed-forward networks struggle with. Unlike a standard feed-forward neural network, which consists of an input layer, two hidden layers, and an output layer, RNNs are designed to operate over sequences of vectors, making them particularly suitable for processing text data and other sequential data [11, 12].

In a feed-forward neural network, each input is individually multiplied by a weight, and the results are aggregated across all the inputs to each node, including a bias term. The total net input for each node is then passed through an activation function to produce a new output, which is subsequently forwarded to the next layer. This process continues until the final output nodes are reached.

RNNs, on the other hand, can process sequences of vectors, allowing them to capture dependencies and patterns within sequential data. This capability makes RNNs advantageous for tasks such as natural language processing, speech recognition, and time series analysis.

Long Short-Term Memory (LSTM) units address the challenge of capturing long-range dependencies in RNNs by incorporating mechanisms to both forget and remember information over time. This is achieved through the addition of an extra context layer within the network known as the cell state, which includes gates that regulate the flow of information into and out of the cell state. An LSTM unit typically consists of three gates: the forget gate, the input gate, and the output gate [11, 12].

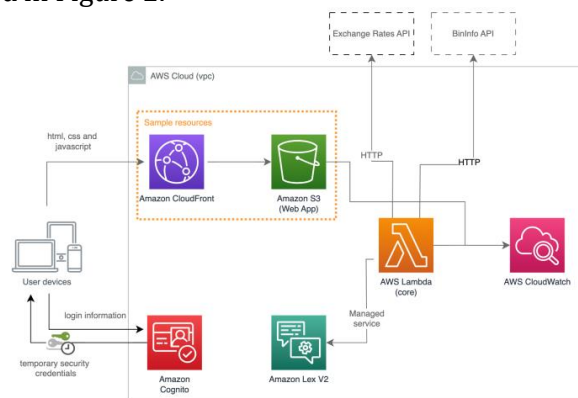
The forget gate controls the removal or "forgetting" of irrelevant information from the cell state. The input gate regulates the addition of new information that is deemed relevant for the current context, while the output gate determines what information is to be output from the cell state. By leveraging these mechanisms, LSTMs are capable of effectively managing and utilizing context over extended sequences, making them particularly effective for tasks involving sequential data such as natural language processing, speech recognition, and time series analysis.

It should also be noted that the most effective architecture for sentiment analysis of text is a recurrent neural network with LSTM blocks. Due to its relatively high accuracy, it enhances the natural language processing recognition process [13].

### **3.1. Description of the tools and architecture of the bot**

In order to build the core of the bot and integrate it, a whole range of AWS services needs to be utilized. These services not only serve as an integration platform, but also act as a tool for

research, as they provide a multitude of functionalities for gathering statistics and analysis. The infrastructure is depicted in Figure 2.



**Figure 2:** The architecture of the serverless chatbot

The integration of the chatbot can be architecturally divided into two blocks, each responsible for a specific part of the infrastructure:

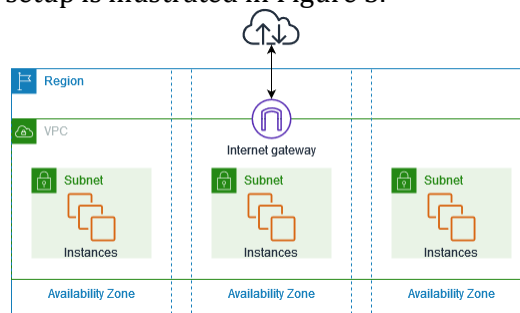
- Web Application – responsible for deploying and maintaining a static web page. All configurations take place via a created template in CloudFormation (configuration template in JSON format). With the help of this template, deploying a custom web page is quite straightforward since lambda functions generate all the necessary resources for the site, and the service creates access to other required services and configures them;
- Lex Bot – responsible for the bot and its configuration. This includes configuration, alias (which is essentially an identifier of the build), version (linked to the alias), and the lambda function that acts as the backend service for the bot.

### 3.2. The setup and deployment of the bot

The setup of the chatbot occurs in several stages.

#### 3.2.1. The network configuration

To enable the bot to call third-party APIs using Lambda and access the Internet, a series of configurations must be set up within the AWS VPC (Virtual Private Cloud) service. This typically involves configuring the VPC, and its network table, and establishing both public and private subnets. This infrastructure setup is illustrated in Figure 3.



**Figure 3:** The VPC structure

All resources and network configurations are built within a specific region. Amazon's cloud computing resources are located in numerous locations worldwide. These locations consist of AWS regions, availability zones, and local zones. Each AWS region is a separate geographic area. Regions have multiple isolated locations known as availability zones. It is most practical to choose the following relationship: region to availability zone:

- Region: Europe (Ireland) or eu-west-1;

- Availability Zone: \*.eu-west-1.amazonaws.com or \*.eu-west-1.api.aws.

In order to access the Internet, it is necessary to separately connect the Internet Gateway to one of the public subnets. The Internet Gateway is a horizontally scalable, redundant, and highly available component of the VPC, serving as the link between the VPC and the internet. It supports both IPv4 and IPv6 traffic and does not create availability risks or impose limitations on network traffic bandwidth [14].

It allows resources in public subnets (Figure 4), such as Lambda functions, to connect to the Internet if the resource has a public IPv4 or IPv6 address. Similarly, Internet resources can initiate connections to services within the subnet using a publicly accessible IPv4 or IPv6 address. For example, the Internet gateway enables connecting to an EC2 instance on AWS from a local computer. It also provides the destination in VPC route tables for traffic destined for the Internet. For IPv4 communication, the Internet gateway also performs Network Address Translation (NAT). This translation is not required for IPv6 communication, as IPv6 addresses are publicly accessible.



**Figure 4:** The diagram of using the Internet Gateway

### 3.2.2. Setting up the chatbot in the Lex V2 environment

Amazon Lex V2 allows the creation of programs using a voice or text interface based on the same technology as Amazon Alexa. Below is the sequence of steps involved in working with Lex:

1. Creating a bot and configuring it with one or multiple intents that need to be supported. The bot is configured to understand the user's intent, engage in a conversation to gather information, and fulfill the user's intent;
2. Testing and exploring the bot. This primarily involves using the test client window provided by the Amazon console;
3. Publishing a version and creating an alias;
4. Deployment of the bot. The bot is deployed on platforms such as mobile apps or social platforms like Messenger and Slack.

When a user makes a specific statement, Amazon Lex uses natural language understanding to comprehend the user's request and returns the most likely intent defined by the bot by default.

In some cases, the bot's linguistic core may struggle to determine the most likely intent. For instance, a user might make an ambiguous statement, or there might be two similar intents. To aid in ascertaining the correct intent, it is necessary to combine domain knowledge with confidence scores from the list of alternative intents. A confidence score is an assessment given by Lex that indicates how confidently the correct intent is identified.

To differentiate between two alternative intents, their confidence scores must be compared. For instance, if one intent has a confidence score of 0.95 and another has a score of 0.65, the first intent is likely correct. However, if one intent has a score of 0.75 and another has a score of 0.72, there is some ambiguity between the two intents, which can be disambiguated using domain knowledge in the program.

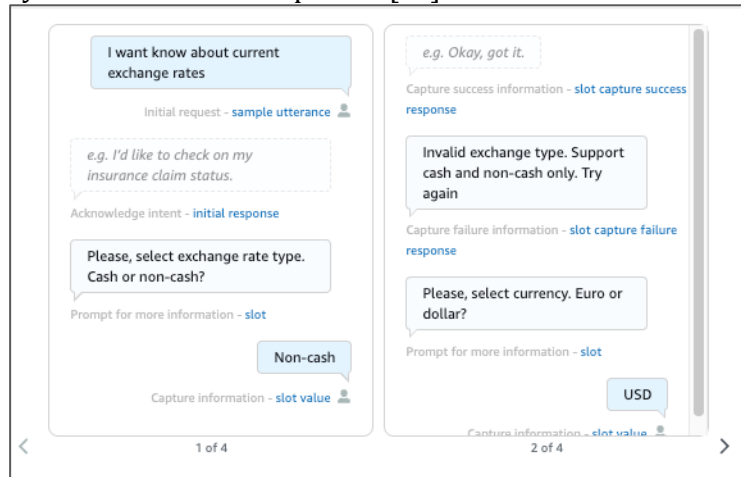
Bot building takes place through Amazon Lex console, the main logic of the created chatbot is a banking assistant. Among the main parameters to mention:

1. Language - English (US);

2. Number of intents – 10. Among the topics the bot can talk about, 6 are primary, i.e., functional, and 4 are more service-oriented, for example, help;
3. Number of slots (types of client data) – 6. Some slots are used several times, but mainly each intent has its own slot.

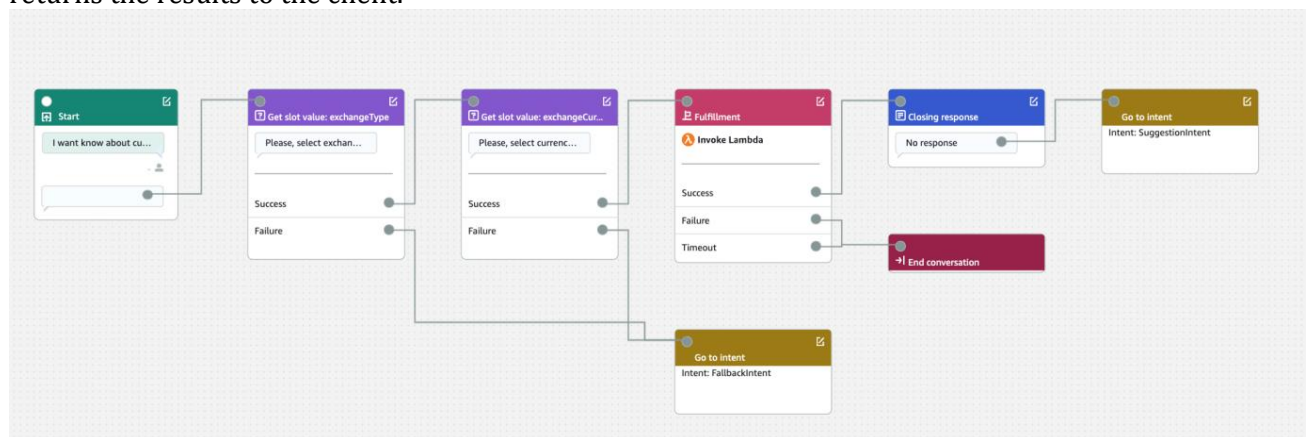
Based on the provided dialogue examples, the bot generates a conversational flow with the user, understanding the context of their intent (see Figure 5). This allows for a more natural and intuitive interaction, enhancing the overall user experience.

Training chatbots using machine learning-based approaches typically requires a vast amount of training data to synthesize suitable responses [15].



**Figure 5:** Example conversation flow for the intent

The structure of all intents is more or less similar (Figure 6), and the user is expected to provide an utterance that initiates the conversation with a specific context. After determining the context, the bot awaits input from the client, sends it to the lambda function for processing, and returns the results to the client.

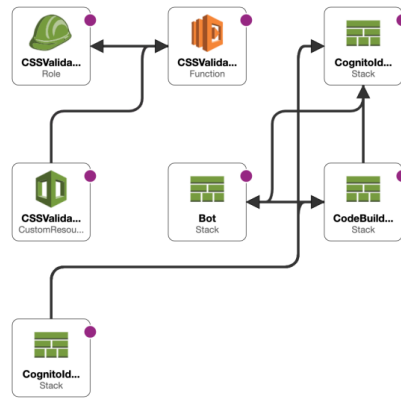


**Figure 6:** Conversation flow chart for intent

### 3.2.3. Setting up a public access point

For setting up the web page and its deployment, the primary service is CloudFormation. The stack schema for the web page is automatically generated by the service (Figure 7).



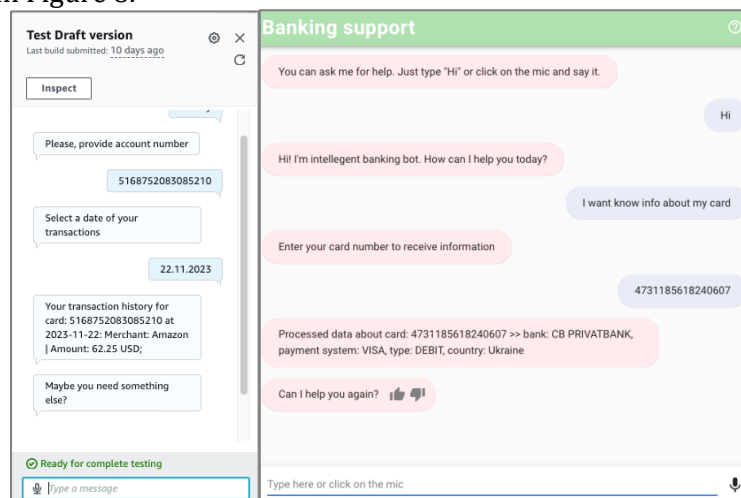


**Figure 7:** Schematic diagram of the main website setup stack in CloudFormation

Basically, for website deployment, need to prepare a template in JSON format that describes the interaction of all necessary services and their dependencies. This will all be automatically generated and configured. By default, the CloudFormation template creates an Amazon Cognito Identity Pool [16]. It also copies the web interface program of the chatbot to an Amazon S3 bucket, including a dynamically created configuration file. The CloudFormation stack provides a link to the demo and the corresponding configuration after deployment.

#### 4. Experiment

Once all the necessary configurations are in place, the bot is ready for testing to ensure its functionality and responsiveness in various scenarios and user interactions. An example of intent testing is depicted in Figure 8.



**Figure 8:** Example of intent testing

Expanding on the topic, when working with Amazon Lex, it's important to understand how the confidence scores for intents are derived. The confidence score represents the likelihood of a particular intent being the correct interpretation of a user's input. The scores range from 0 to 1, where a score of 1 indicates high confidence in the intent being accurate.

These confidence scores are used to assess the accuracy and reliability of the bot's interpretations. By utilizing these scores, developers can create test programs to evaluate the impact of changes in intent expressions on the bot's behavior. By collecting confidence scores for different intent expressions and then updating the intents with new expressions, one can measure the effectiveness of these changes and their impact on the bot's performance.

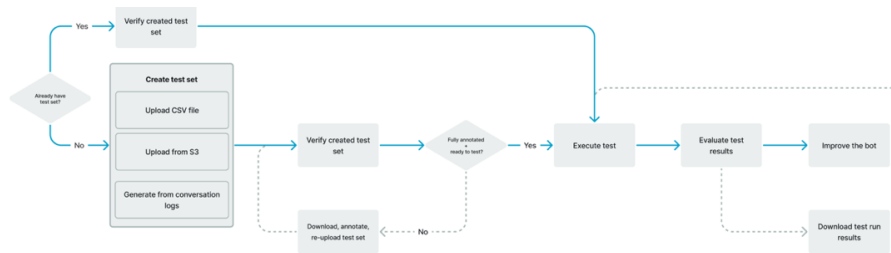
Note that confidence scores are comparative and should not be regarded as absolute measures of correctness. They can fluctuate based on the bot's improvements and adjustments.



Furthermore, when Amazon Lex processes user requests, it not only identifies the most likely intent but also provides up to four alternative intents along with their corresponding confidence scores. This information is crucial for understanding how well the bot is interpreting user inputs and can aid in refining and optimizing its performance.

For automated testing of an Amazon Lex chatbot, Test Workbench (Figure 9) may not be the ideal tool. Instead, it's recommended to utilize specialized testing tools and frameworks tailored for Amazon Lex, such as AWS SDK, AWS CLI, or other AWS testing tools, which provide the capability to automate interactions with the bot, execute tests on specific bot responses, and set up test scenarios to ensure the bot functions as expected in various use cases [17].

By using these tools and frameworks, developers can automate the process of sending requests to the chatbot, validating the responses, and testing its behavior under different circumstances. This approach not only ensures the reliability and accuracy of the bot's responses but also streamlines the testing process, making it easier to identify and address any issues that may arise.



**Figure 9:** Test workbench settings

## 5. Results

Following the chatbot testing, the outcomes have been compiled into Table 1.

**Table 1**  
The results of the chatbot research

Intent	Type	Confidence score									
		1	2	3	4	5	6	7	8	9	10
Exchange rates	Text	0,97	0,97	0,96	1	0,55	0,96	0,78	0,91	0,62	0,86
	Voice	0,94	0,96	0,94	1	0	0,96	0,79	0,91	0,62	0,73
Customer data	Text	0,82	0,94	0	0,93	0,9	0,85	0,75	0,55	0,92	0,93
	Voice	0,86	0,76	0	0,79	0,81	0,78	0,75	0,54	0,77	0,9
Transaction history	Text	0,83	0,7	1	0,93	0,73	0,62	0,89	0,79	0,9	0,84
	Voice	0,73	0,7	0,94	0,79	0,51	0,59	0,89	0,61	1	0,83
Card check	Text	1	0,91	1	1	0,95	1	1	0,91	0,76	0,64
	Voice	1	0,84	1	1	0,94	0,98	1	0,89	0,74	0,58
Banking support	Text	1	1	0,98	1	1	0,72	0,92	0	0,79	0,91
	Voice	1	1	0,92	0,98	0,94	0,65	0,88	0	0,63	0,61
Card balance	Text	0,96	1	0,96	0,95	0,89	1	0,88	0,62	0,84	0,56
	Voice	0,9	1	0,92	0,93	0,8	0,89	0,82	0,42	0,56	0,6

To conduct the experiment, a dataset of possible user statements for each intent was prepared in advance. These statements could trigger the execution of the respective intentions. After running the tests manually through the Amazon console, the test results could be used for automated testing. Testing was carried out on intents simulating certain banking services, such as checking card balance, transaction history etc.

The Test Workbench tool is used to create a test suite for automating the testing of each intent. With an average of 10 test runs for each intent (during the study, there were 6 of them), each containing up to 5 utterances for recognition, the following results can be obtained: the accuracy of recognizing textual input ranges from 60% to 99%, and voice input is generally 10% less accurate, depending on the clarity and pronunciation level of the English speaker making the request.

Post-testing intents statistics were compiled based on 734 recognized utterances. This means that the bot comprehended this number of user input conversations and successfully interpreted them into 519 intents. However, there were 159 other utterances that, although recognized, were not mapped to any specific intents.

The obtained results point out certain shortcomings of the service:

- For higher accuracy, more utterances are needed. Since only five example utterances have been created for each intent that a user can input, the recognition accuracy is not as high, and in some cases, it is zero when there are no keywords. As the number of utterances is limited to 100 units, this may cause issues when creating a more complex bot;
- Voice input performs less effectively. The results indicate that tests involving voice input have a lower reliability score, even when the bot accurately recognizes the text. This can be explained by the fact that voice interpretation into text does not consider punctuation (in cases where the utterance matches the test input). Additionally, the user's accent can also influence the score, as the bot interprets some input completely differently from how the user spoke.

Other, more global problems arise from the limitations imposed by the AWS Lex NLP Engine on each individual bot. Even though the testing did not reach the limit during the study, the Amazon NLP mechanism only allows for 100 intents per bot, meaning that a bot can handle only 100 different queries. However, does 100 intents provide a sufficient quantity for an enterprise application?

The first issue is that the limit of 100 intents includes auxiliary (service) topics, which makes the conversation less robotic. To make conversational bots more human-like, they also need to handle elements such as greetings and small talk, essentially being able to respond to phrases like "How are you today?", "What time is it?", "What's the weather like?", "Who created you?", etc.

A typical conversation can involve 20, 30, or even 50 intents, as developers strive to account for all the things users ask digital AI assistants. They often express dissatisfaction when a chatbot cannot handle these simple phrases effectively. Having 20 representative, nonfabricated examples per intent is a lot for creating a new conversational AI [18].

The second issue arises with more complex business processes such as order processing, refund management, or complaint resolution, which often involve numerous permutations around what a client may ask or with what a process may be associated. This complexity diminishes the number of remaining intents available, as developers account for variations and intricacy.

In this context, perhaps the solution lies in the architecture. If creating a bot that can provide a good experience requires 700 intents, then at least 7 Lex bots working together, each handling different conversation segments, might be necessary to bypass this limitation. This can be achieved using the Lex Network of Bots.

The Network of Bots provides a unified working experience for multiple bots. It allows the addition of several bots to one network to ensure flexible and independent bot lifecycle management. The network offers end users a single unified interface and directs the request to the appropriate bot based on the user's input [19].

As chatbots are continuously being enhanced, they may still be susceptible to functional failures [20].

## 6. Discussions

Amazon Lex proves remarkably versatile across diverse domains, as its efficacy hinges on finely-tuned intent configurations tailored to user interactions. This chatbot finds seamless integration

potential in fields ranging from healthcare to military applications, serving as a voice-driven assistant adept at diverse tasks.

The test trials conducted on the developed chatbot revealed a recognition accuracy ranging from 60% to 99% for text inputs and 50% to 90% for voice inputs across intents containing up to five statements. These findings underscore the influence of clarity and pronunciation levels on linguistic recognition.

Looking ahead, the scope for future research in chatbot technology is extensive, holding promise for transformative impacts on customer service, sales, and internal business operations. By automating mundane tasks and addressing common inquiries, chatbots liberate human agents to tackle more intricate queries, ultimately enhancing operational effectiveness.

To improve the overall level of recognition and performance, it is worth considering the integration of additional services from AWS:

- **Amazon Comprehend:** This natural language processing (NLP) service can analyze text for sentiment, entities, key phrases, and language detection. Integrating Amazon Comprehend with Amazon Lex allows your chatbots to understand user intents more accurately and extract valuable insights from user inputs. For example, Comprehend can identify specific entities mentioned by users, such as product names or locations, enabling bot to provide more personalized responses [21];
- **Amazon Polly:** Amazon Polly is a text-to-speech (TTS) service that can convert text into lifelike speech in various languages. By integrating Amazon Polly with Amazon Lex, developer can enhance the user experience by enabling your chatbots to respond to user queries with natural-sounding speech [22].

To address issues underlying Lex, such as limitations on the number of utterances, it is necessary to resort to using services like Network of Bots.

When comparing the performance with other alternatives, one can refer to the article «Building Chatbot Using Amazon Lex and Integrating with A Chat Application» [23]. Following the bot integration, the authors reached a similar conclusion that enhancing the overall recognition quality, particularly in voice recognition, requires increasing the number of utterances and utilizing more AWS services.

## 7. Conclusions

The developed chatbot elevates the user experience while engaging with voice consultants by offering flexible customization options.

The research concluded that Amazon Lex, despite its extensive array of features, encounters inherent restrictions, particularly concerning its linguistic capabilities. Platform's support for a restricted range of languages could serve as a hindrance for organizations and developers seeking to implement conversational AI solutions in multilingual environments. Nonetheless, this obstacle can be effectively circumvented by harnessing additional Amazon services to craft customized bots from the ground up, thereby bolstering language inclusivity and diversification.

Another critical area of limitation lies in the constraints related to intents. While these limitations vary across different NLP engines, it has been observed that some competing platforms offer more robust intent capabilities compared to AWS NLP. Despite this, the orchestration benefits provided by the Network of Bots feature within Amazon Lex persist as a main advantage. By employing a microservices architecture, developers can solve the challenge of intent limitations, creating more comprehensive and effective conversational AI solutions. This approach also facilitates the integration of various NLP mechanisms to overcome additional constraints, such as natural language understanding and slot filling.

Throughout the development process the web application was designed, featuring interactive text input fields and voice recording functionalities. The backend architecture of the application was intricately configured to orchestrate the seamless transmission of data through the AWS SDK to Amazon Lex. Here, the information undergoes processing, culminating in the generation of responses that are dynamically showcased on the web page. This integration underscores the

synergy between the user interface and Amazon Lex, fostering an increased user experience through the adept utilization of both voice and text inputs.

The flexible architecture of multi-bot systems enabled by the Network of Bots features not only offers a solution to intent limitations but also presents a myriad of deployment possibilities. This structural design allows for the amalgamation of various conversational AI solutions, providing organizations with the flexibility to capitalize on the technology by deploying it in diverse scenarios and environments.

The conducted test runs of the developed chatbot, for each intent containing up to 5 statements for recognition, showed a text input recognition accuracy of 60-99%, and voice input recognition accuracy of 50-90%. This indicates that linguistic recognition depends on the clarity and the pronunciation level of the speaker.

The future research directions for chatbot technology are vast: they have the potential to revolutionize customer service, sales, and even internal business processes. By automating routine tasks and handling frequently asked questions, chatbots free up human agents to focus on more complex queries, thus improving operational efficiency.

## 8. References

- [1] J. Fu, S. Mouakket, Y. Sun, The role of chatbots' human-like characteristics in online shopping, *Electronic Commerce Research and Applications*, 2023.
- [2] C. Wang, Y. Li, W. Fu, Whether to trust chatbots: Applying the event-related approach to understand consumers' emotional experiences in interactions with chatbots in e-commerce, *Journal of Retailing and Consumer Services*, 2023.
- [3] V. Vijayaraghavan, J. B. Cooper, R. Leevinson J., Algorithm Inspection for Chatbot Performance Evaluation, *Procedia Computer Science*, 2020.
- [4] S. Sahai, Explain the key features of Amazon Lex, 2022. URL: <https://www.projectpro.io/recipes/explain-features-of-amazon-lex>.
- [5] C. Greyling, A Critical Assessment Of The Gartner Magic Quadrant for Conversational AI Platforms, 2022. URL: <https://cobusgreyling.medium.com/a-critical-assessment-of-the-gartner-magic-quadrant-for-conversational-ai-platforms-6ac538b5fc8c>.
- [6] S. Singh, H. Beniwal, A survey on near-human conversational agents, *Journal of King Saud University - Computer and Information Sciences*, 2022.
- [7] Dev.to community, From Amazon Lex to GPT-4: How to make a bot with your own data, 2023. URL: [https://dev.to/vladimir\\_mvsv/from-amazon-lex-to-gpt-4-how-to-make-a-bot-with-your-own-data-3cih](https://dev.to/vladimir_mvsv/from-amazon-lex-to-gpt-4-how-to-make-a-bot-with-your-own-data-3cih).
- [8] F. Mansoor, Predict Potential Diseases with Amazon Lex Chatbot, 2024. URL: <https://www.toolify.ai/ai-news/predict-potential-diseases-with-amazon-lex-chatbot-1350740>.
- [9] D. Scharp, M. Hobensack, A. Davoudi, Natural Language Processing Applied to Clinical Documentation in Post-acute Care Settings: A Scoping Review, *Jamda*, 2024.
- [10] C. V. Misischia, F. Poetze, C. Strauss, Chatbots in customer service: Their relevance and impact on service quality, *Procedia Computer Science*, 2022.
- [11] M. McTear, Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots, in: *Synthesis lectures on human language technologies*, Morgan & Claypool Publishers, Toronto, Canada, 2021. doi:10.2200/S01060ED1V01Y202010HLT048.
- [12] P. Choudhary, S. Chauhan, An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism, *Decision Analytics Journal*, 2023.
- [13] K. Ostrovska, I. Stovpchenko, D. Pechenyi, Research of methods based on neural networks for the analysis of the tonality of the corps of the texts, *System Technologies. Regional interuniversity collection of scientific papers*, Ed. 4, Dnipro, 2023, pp. 155-167. doi:10.34185/1562-9945-4-147-2023-14.

- [14] Amazon Documentation, Connect to the internet using an internet gateway, 2024. URL: [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_Internet\\_Gateway.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html).
- [15] C. Kecht, A. Egger, W. Kratsch, Maximilian Röglinger, Quantifying chatbots' ability to learn business processes, *Information Systems*, 2023.
- [16] Amazon Documentation, What is Cloud Formation. Detailed documentation, 2024. URL: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html>.
- [17] Amazon Web Services, Evaluating bot performance with the Test Workbench, 2024. URL: <https://docs.aws.amazon.com/lexv2/latest/dg/test-workbench.html>.
- [18] A. R Freed, *ConversationalAI. Chatbots that work*, Manning Publications Co, 2021.
- [19] Amazon Web Services, Creating a network of bots for Amazon Lex V2, 2024. URL: <https://docs.aws.amazon.com/lexv2/latest/dg/network-of-bots.html>.
- [20] A. Janssen, D. R. Cordona, J. Passlick, M. H. Breitner, How to Make chatbots productive – A user-oriented implementation framework, *International Journal of Human-Computer Studies*, 2022.
- [21] Amazon Documentation, What is Amazon Comprehend, 2024. URL: <https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html>.
- [22] Amazon Documentation, Amazon Polly detailed guide, 2024. URL: <https://docs.aws.amazon.com/polly/latest/dg/what-is.html>.
- [23] A. S. S. K. Sreeharsha, S. M. Kesapragada, S. P. Chalamalasetty, Building Chatbot Using Amazon Lex and Integrating with A Chat Application, *International Journal of Scientific Research in Engineering and Management*, 2022. doi: 10.55041/IJSREM12145.