A Comparative Evaluation of Chatbot Development Platforms

Ioannis Dagkoulis
Department of Computer Science, International Hellenic University, Kavala, East Macedonia & Thrace, Greece
iodagko@cs.ihu.gr

Lefteris Moussiades
Department of Computer Science, International Hellenic University, Kavala, East Macedonia & Thrace, Greece
lmous@cs.ihu.gr

ABSTRACT

Chatbots and virtual assistants have become part of people's everyday life. The need for mass production of these services rapidly and efficiently has created an explosion of software-related services focused on developing chatbots. Big companies like Google, Microsoft, Amazon, and IBM offer complete Chatbot Development Platforms and compete with each other. Our effort is to help people interested in using these platforms decide which is the best CDP for their case. Similar attempts have happened but are now outdated as CDPs have introduced breaking changes. We study each CDP, define criteria and calculate scores based on requirement assumptions. In parallel, we observe how innovations in NLP are presented in the market through CDPs.

CCS CONCEPTS

• **Development frameworks and environments**;

ACM Reference Format:
Ioannis Dagkoulis and Lefteris Moussiades. 2022. A Comparative Evaluation of Chatbot Development Platforms. In *26th Pan-Hellenic Conference on Informatics (PCI 2022), November 25–27, 2022, Athens, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3575879.3576012

1 INTRODUCTION

Nowadays, people make use of AI applications in the form of chatbots and personal assistants. Although chatbots and personal assistants differ in their goals, they share the same technology. For example, a chatbot can contribute to increasing the productivity of a customer support department, while well-known virtual assistants such as Alexa, Google Assistant and Microsoft Cortana act as personal assistants helping a person in their daily activities

But what technology supports the beautiful possibility of human-machine communication with natural language? The core of this technology is referred to as Natural Language Processing (NLP). The first steps in using NLP take us back to 1950 [1] when Alan Turing proposed a program such that when communicating with users, the latter do not realize that they are talking to software but think they are talking to a human. Alan Turing defined a test based on this assumption, now called the Turing test. Many different approaches have been performed to resolve the Turing test. Today we have reached a point where it is often required for a chatbot to introduce itself to humans as a bot before starting a conversation. Apart from supporting the functionality of a chatbot or virtual assistant, NLP is used in many other fields like speech recognition, machine translation, text mining, text classification and sentiment analysis.

The need for using chatbots and the corresponding NLP tech has created a new software industry specialized in creating chatbots. Many new software companies have been developed, offering different solutions. Big companies like Google, Amazon, Microsoft, IBM, and Facebook offer complete Chatbot Development Platforms (CDPs) as their cloud infrastructure. Smaller companies are either taking advantage of those platforms, acting as intermediates helping other customers use chatbots in their services or using a CDP for their own needs. In general, a newcomer company in the chatbot industry has difficulty deciding which platform is more suitable for them. Even though there is previous work on this matter, breaking changes have been introduced from CDPs since the latest works. Our objective is to define standard evaluation criteria to evaluate CDPs to reduce the effort for candidates to select the appropriate platform according to their requirements based on the latest versions of CDPs.

The platforms that are going to be evaluated are the following in alphabetical order:

Amazon LEX, Dialogflow CX, ES (Google), LUIS, CLU (Microsoft), RASA, Watson Assistant (IBM)

These CDPs were chosen because they offer a complete set of tools to create a chatbot for every use case scenario. All offer an NLU unit as a service and follow similar system architecture. An exception is RASA which is a slightly different case as it is an open-source solution offered by a smaller company in terms of resources. However, it follows a similar architecture and offers most of the tools as the other CDPs.

The rest of the paper is structured as follows. In section 2, we discuss a chatbot platform's architecture in detail. In section 3, we analyze the fundamental concepts of chatbots to understand current and future trends in the field. Next, in section 4, we examine related works to utilize them. In section 5 we analyse the descriptive evaluation method and define the criteria and assumptions while describing the procedure. Finally, section 6 presents our results by providing a comparative table.

2 ARCHITECTURE

CDPs offer a component-based structure that supports their goals' required flexibility and scalability. These components and services are more or less familiar to each CDP, sometimes under different names, and are as follows:

- **NLU service:** NLU is a vital component of a CDP [2], as it is where the effort to understand the user's phrases is made. The overall performance of a chatbot is closely related to the NLU.

PCI 2022, November 25–27, 2022, Athens, Greece

ACM ISBN 978-1-4503-9854-1/22/11…$15.00
https://doi.org/10.1145/3575879.3576012

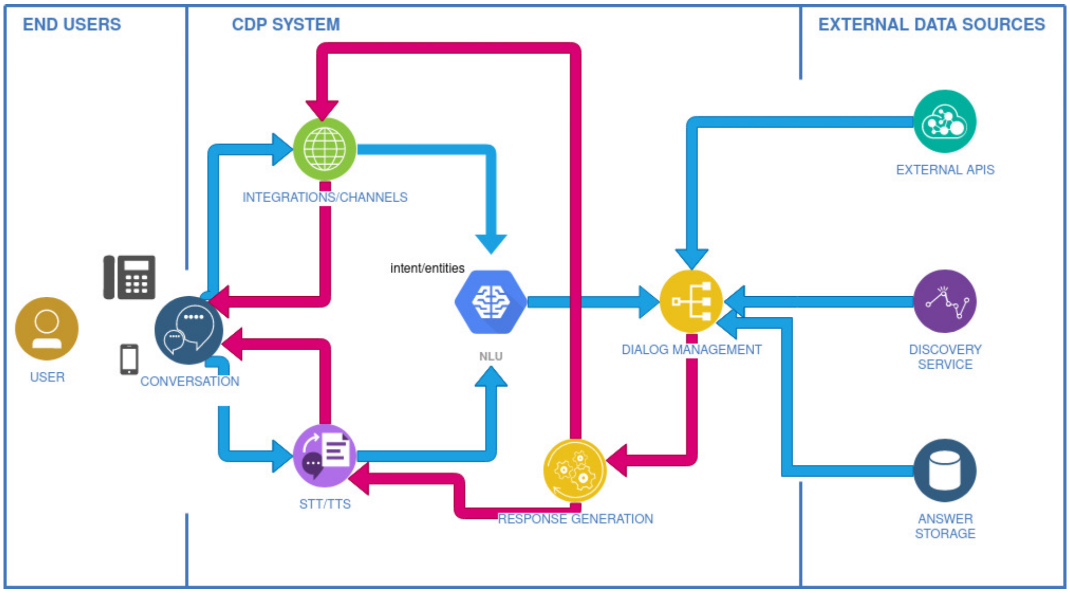

Figure 1: A conceptual representation of a typical CDP workflow

- **Integrations-Channels**: This component helps developers integrate their chatbots with various communication channels. These could be over telephone or text messengers like Signal, Telegram, FB Messenger, Skype etc. or even CRM systems like Salesforce. Most of the time, these integrations are ready-made software clients communicating with NLU's API.
- **Dialogue Manager**: This component handles the dialogue flow of a chatbot. Usually is supported through a graphical interface to visualize the flow. It is closely related to NLU and integrates NLU's concepts like intents, entities, slots etc. Here, chatbot developers coordinate the other components to bring life to the chatbot by building responses from utterances, combining data from webhooks, declaring integrations, etc.
- **Webhooks**: As a general term, it is a method of augmenting or altering the behaviour of a web page or web application with custom callbacks [3]. In a chatbot's context, webhooks receive data from an external service or application during a conversation [4].
- **Response Generator:** This component is usually integrated with the dialogue system. Most goal-oriented chatbot applications generally enrich the response with external data sources
- **Speech to Text and Text to Speech**: These two components usually exist as standalone services and are a common but not integral component of CDPs.

A CDP is designed to support different use case scenarios. The most common is the one illustrated in figure 1. Advanced scenarios include:

- Use of multiple chatbots in collaboration with each other
- Implementation of a chatbot with the use of a visual dialogue flow management system together with code libraries (SDKs) which are used to handle exceptional use cases

3 ESSENTIAL CONCEPTS AND FUTURE TRENDS

Different strategies have been developed to overcome NLP barriers. Those strategies depend on various factors like how NLP fits business requirements, what techniques are used to achieve the goals, the nature of the knowledge domain and several others. Adamopoulou and Moussiades [1] name such factors as knowledge domain (generic, open domain, closed domain), goals (informative, chat-based, goal-oriented or task-based) and response method generation (rule-based, retrieval based, generative). CDPs are mainly focused on goal-oriented services. Two concepts play a significant role in a goal-oriented NLU**, intent and entity**. For each user's utterance chatbot tries to understand the user's intent and labels the utterance with the intent. NER (Named Entity Recognition) is a closely related concept affecting intent identification. Named Entities are words or phrases that serve as instances of an entity. **Slot filling**: Slot filling means that chatbot has to gather all required information arguments from the user to fulfil a response. For example, when a user wants to book a flight (intent), they must define when, from, to arguments for the chatbot to accomplish the goal of securing a getaway. Note that Named Entities are usually the best candidate for filling these slots. Slot filling with named entities is closely related to intent classification, and a joint architecture is proposed that should be used for slot filling and intent prediction [5]. **Context**: Another critical factor is the context of the conversation. It is usually preserved as a conversation state during the dialogue between the chatbot and the user. It comprises elements of the dialogue that preceded, possibly some historical features or, depending on the case and other data sources. It also facilitates the reliable identification of the user's intention. The chatbot must maintain the context of discussion until it fulfils its goal. **NLU component**: The NLU element performs the actual intent recognition by exploiting the filled slots. It is based on Machine Learning (ML) techniques. It is reasonable that CDPs try to implement state-of-the-art ML techniques to achieve better performance. Nowadays, state-of-the-art is considered contextual language pre-trained models using transformers and attention mechanisms.

Such a well-known model is BERT [5] which can be trained either as feature-based or fine-tuned. Unfortunately, apart from RASA [6], which is open-sourced and lets the developer decide which ML techniques they want to use, the other commercial CDPs do not always reveal their NLU implementation. In their work [7], there is a comparison between BERT and commercial CDPs on intent detection. The writers point out that commercial CDPs consider three other vital factors apart from achieving the best accuracy in benchmarking tests. Training data limitations, non-standard user input robustness, and computational efficiency. For example, Rasa created its model DIET [8], inspired by pre-trained language models like BERT, focusing on computational efficiency instead. **Dialog Manager:** [9] coordinates the operation of the abovementioned elements. It contributes to the conversation's continuity, managing simultaneous intentions, recording follow-up intents to help train bots, etc. A Dialog manager typically takes the form of a decision tree or state machine.

In future trends, we should mention **End-to-End Dialogue Systems.** These systems bypass the modular architecture of traditional CDPs and utilize a language model trained to read utterances as input and output responses. Alexa Conversations [10] is such an attempt to implement an end-to-end dialogue system. In contrast with conventional modular systems, there are benefits like no need for intent classification, preserving a dialogue policy or context state maintenance. The drawbacks are that they need much more training data, and the results could be inconsistent and less explainable regarding debugging errors

4 RELATED WORKS

Two methods are being used to compare CDPs, descriptive and performance-based. Each method is focused on different aspects of a CDP and is needed to have a thorough overall evaluation. The performance-based method mainly involves the NLU. The key factors to perform a comparison against are the following:

- Accuracy of intent and entity prediction
- Amount of training time. In real-life scenarios, the time needed to train a chatbot affects the overall performance. IBM researchers [7] have noted in their work that there is a trade-off between training time and accuracy in commercial CDPs.
- Datasets close to real-world data. Real-world datasets usually contain a small amount of data that could be either too generic or too specific on the same domain, imbalanced data distribution etc.

Arora et al. [11] performed tests on CDPs (Rasa, Dialogflow, LUIS included) using newly created datasets close to real-world cases. Their tests show that CDPs fail to achieve scores like 90+ as in previous benchmarks with artificial datasets. Furthermore, Larson et al., in their work [12], show that incorporating out-of-scope data in training is vital in improving the model's out-of-scope performance. Finally, as the performance method could be automated, there are some attempts in this direction to evaluate chatbots like ChatEval [13]. Also, ParlAI [14] could serve as a primitive example of this approach. Using the descriptive method, we first need to define specific criteria that will help us categorize the CDPs. Their work [15] defined 34 criteria in 10 different domains. They approach the problem from a technical and managerial perspective. Another attempt is [16], where they use eight criteria to evaluate the CDPs based on a business perspective. They innovate using the Analytical Hierarchy Process (AHP) tool for decision-making. Also, the work [17] uses pros and cons as an extra feature apart from the criteria. Finally, their work [19] defines six different categories with 15 criteria in total

5 DESCRIPTIVE EVALUATION

In descriptive evaluation, the method is to define the characteristics of a CDP and compare these with the competitors. In their work, Perez and Soler [15] classify their criteria into two factors: managerial and technical. Kostelník et al. [16] base their criteria on CDP characteristics to decide which CDP fits better for a small or big company. Finally, Braun and Mathes [18] defined requirements as criteria and compared them with each CDP's capabilities. Our criteria are a mixture of shared characteristics and requirements in today's CDPs. Score criteria presented in Table 1 have a value range between 1 and 4. For each criterion, the highest score is awarded to the best candidate, as this is a comparative procedure

Visual management of dialog flow: It is evident in the latest versions of CDPs an attempt to build chatbots with no code. For example, Dialogflow created the CX edition, which enrols a visualized state machine that manages dialogue flow called the visual builder. On the same path, Microsoft made Bot Framework Composer, a visual authoring tool for building conversational AI applications. Moreover, Microsoft created another service called Power Virtual Agents which attempts to implement a no-code solution. Watson's assistant followed a visual dialog approach using dialogues from the start. In 2020 introduced actions which simplify some everyday use case scenarios on building a chatbot. Amazon Lex v2 presented a conversation flow which visualizes a conversation between a user and a bot during development. Also, Amazon has created Genesys Cloud, a no-code solution like Power Virtual Agents of Microsoft, which contains a dialog engine bot flows visual management service. Lastly, Rasa does not support any visual dialog management flow.

Prebuilt agents: Prebuilt agents have already created working examples of chatbots in a specific domain. Dialogflow offers about 50 prebuild agents in DialogFlow ES and 9 in Dialogflow CX; Lex offers 3, Watson only one, Microsoft offers 7, and Rasa does not provide any. An exciting feature that is in preview status is Automated Chatbot Designer by Amazon Lex. The idea is to produce intents and slots from existing conversation transcripts. Watson Assistant in the Plus plan supports the same idea

Integration channels: The chatbot needs to reach as many audiences as possible through communication channels. CDPs offer

Table 1: Definition of score criteria

Score	Description (this description does not always fit with the characteristics)
1	The characteristic is not supported at all
2	The characteristic exists as functionality but misses critical parts
3	The characteristic is merely supported
4	The characteristic is fully implemented

some ready-made integrations. These include text-based (messenger, Twitter etc.) or voice-based media (telephony). Microsoft provides this capability through azure connections (currently supports 13 channels). Dialogflow supports three telephony channels and three text-based channels. Watson Assistant offers five ready-made integration channels. Amazon Lex supports out-of-the-box Facebook messenger, Slack and Twilio. It also provides integrations to contact centres like Amazon Connect Omni channel or Amazon Genesys Cloud. Finally, Rasa supports ten-channel integrations.

Search knowledge service: This service retrieves info from documents, web content and other existing knowledge management tools. Currently, Watson Assistant offers a search skill, a particular skill that collaborates with Watson Discovery (search service). Dialogflow uses knowledge connectors in CX, which take advantage of the Google search indexer for public data or uses an experimental QA service for private data. Microsoft offers a new tool called Azure Cognitive Service for Language, a substitution for three different services, LUIS, text analytics and QnA maker. This tool implements a question-answering service which acts as a search service. Amazon Lex offers Amazon Kendra as a search service which can be combined with intent. Amazon also offers two other services, Amazon Textract and Amazon Comprehend, which could be combined with lambda functions. These services can be used through intent fulfilment. Rasa could use a custom action and call a search service from other cloud-based CDPs.

Language support: Dialogflow supports 123 languages. Amazon Lex supports only 18 languages. Both CDPs add new languages to the same bot. On the other hand, Watson Assistant only lets users specify the language per assistant, so multiple assistants need to be created for each language. Watson supports 13 languages and a universal one. IBM claims that universal language can be trained on specific language utterances and then operate on this particular language. Microsoft supports multiple languages using a universal model [19]. In general, it works out of the box when adding new languages without the need to add additional examples in those languages. Rasa supports one language per assistant, and one can use either pre-trained models or word representations in this specific language.

Development Adaptability: Building a chatbot may require a different approach based on the requirements. Both high-level visual management of dialogue flow and code-written functionality may be needed to accomplish a state-of-the-art chatbot. Microsoft offers various tools which could be used in different scenarios or even in combination, like bot framework SDK, bot framework composer and Virtual Power Agents. Watson Assistant permits code developers to interfere when needed. This is possible in actions, webhooks and channels. DialogFlow, in general, follows the same approach as Watson Assistant. As an extra, it offers an inline editor for helping write rapidly new webhooks called fulfillments (fulfillments have a more general meaning in the CX version). Amazon Lex provides the same functionality as Watson Assistant. Rasa does not offer visual management of dialog flow. Instead, it offers stories, a text-based tool for designing dialog flows. Due to its open-source nature is the most configurable and extensible among the CDPs. Still, at the same time, Rasa offers the Rasa Enterprise version, where the developer gets additional services out of the box and support.

Multiple agents architecture: Dialogflow uses multiple agents collaboration in CX and ES but differently in each version. ES uses a Mega agent as a broker sending the utterances to subagents that fulfil a response. In CX, flows are used instead. This functionality is required for complex scenarios and is used to divide a project into smaller parts worked by dedicated teams. Even though Lex offers the possibility to create multiple assistants or bots, it does not provide any collaboration between agents. Microsoft utilizes a bot orchestrator, which substitutes the bot framework dispatcher. Its main job is to route user utterances to the appropriate skill, which acts as an internal bot. The orchestrator decides which skill should handle the user's utterance based on deep learning ML techniques.

Moreover, Microsoft supports bot components directly analogous to a shared code library. IBM Watson Assistant uses the concept of skills as Microsoft so that each assistant can have more than one skill. Rasa does not explicitly offer a multiagent architecture, but it is possible to achieve similar results by using the programming language capabilities of modular programming

Life Cycle Management: Developing a chatbot is an iterative workflow. This workflow usually consists of a development phase, a testing phase and a production one. DialogFlow, especially in the CX edition, provides specific tools to support this workflow. An agent can be in different environments and versions. The training can be automatic or manual and depends on the NLU type (standard/advanced). Amazon Lex uses versions and aliases to handle the development state of the chatbot. It's worth noting that not only the bot can have versions but also intents and slot types**.** IBM Watson Assistant supports two environments, draft and live. All content can be developed in either environment, including the development of channels. In Microsoft's CDP, when the developer uses either bot framework Composer or bot framework SDK in collaboration with an NLU like LUIS or Cognitive Services for Language, they can use a code versioning system like GitHub to handle the life cycle of the bot. LUIS uses a versioning system for supporting different versions of a model, while Cognitive Services for Language use deployments instead. Rasa, an open-source solution that can run on-premises, takes advantage of versioning systems (GitHub) and CI/CD tools.

There is also Rasa Enterprise if no resources are available to work on-premises

Testing: Chatbot testing is mainly focused on the NLU component. The usual method is to let the developer create the utterances through a web chat or a CLI (command line interface) and check the chatbot's responses. Rasa provides the ability to write test stories, which are ready-made conversations between a chatbot and a user. Dialogflow CX offers the same functionality with the use of a simulator. Furthermore, there is the possibility of performing automated tests in a specific environment. Also, another feature called Experiments lets developers test different versions of a chatbot against a controlled one in a live setting. Amazon Lex, Watson Assistant and Microsoft CDP offer the usual method described.

Debugging: In the case of chatbot development, it means reading conversations between chatbot and user that happened in the past and trying to find bugs or propose improvements. Bugs are considered, for example, a user abandoning a conversation before reaching the end or having the chatbot stuck asking the same question many times etc. Microsoft offers the possibility to save the conversations as transcript files for studying later. The data are processed with an analytics tool, which helps to visualize aggregated data like retention, users/channels use, activities/channels relation in time etc. Watson Assistant offers analytics for users, conversations, and requests.

Furthermore shows the average completion of conversations and intent recognition. Also, it allows checking conversations that happened in the past with filters (for example, intent recognition), helping the developer discover issues with conversations. DialogFlow offers analytics in their CX edition, presenting interactions and sessions. Also, it provides a History feature that identifies issues with intent matching or webhook errors. Finally, it offers a validation feature which checks the training data and the quality of the page-based flow structure. Amazon Lex provides conversation logs as a tool for debugging conversations. Amazon Lex cooperates with Cloudtrail and Cloudwatch but does not offer customized analytics for the chatbot. Rasa supports the concept of CDD (Conversation Driven Development). Part of it is the review of conversations using Rasa X, a tool for CDD. There is also an experimental feature called markers which marks points of interest in dialogues for evaluation. Finally, Rasa offers analytics only in its Rasa Enterprise version.

Vendor lock-in: Avoid vendor lock-in is always desirable from the consumer's perspective of a product or service. Rasa is the best option in this case, as it is under an open-source license. It is worth mentioning that Rasa X is not open-sourced but free of charge. Rasa has a migration guide which showcases how one could migrate from Dialogflow or LUIS to Rasa. All other CDPs offer the possibility to import/export chatbot data (intents, entities, settings) in JSON format.

Run on-premises: Rasa is the only DCP that can run on-premises entirely due to its open-source nature. Nevertheless, Microsoft Cognitive Service for Language offers the option to run entirely on premises in case there are security or data governance requirements using docker containers. Watson Assistant supports running on-premises in the Enterprise version

Speech-to-text (STT) and text-to-speech (TTS): Voice utterance recognition requires different language understanding models than text recognition ones. Generally, every CDP we check can cooperate with a service that offers STT and TTS services through webhooks. Microsoft offers Direct Line Speech, an STT and TTS service. Dialogflow incorporates its own STT and TTS mechanism; alternatively, one could integrate Dialogflow with Google Cloud STT and TTS. Watson Assistant offers SST and TTS support when selecting integration with the phone, which is an option in the Plus plan. Amazon Lex's bot can directly consume audio files through rest API. Rasa does not offer such a service and needs to use its connector library to take advantage of STT and TTS external providers.

Context Management: In a conversation between people, context understanding is crucial; the same applies to chatbots. CDPs preserve context in memory during conversations using techniques like slot filling, session data handling, state management and ML language models. Microsoft CDP handles context as a conversation state. The state is preserved through code by offering ready-made code libraries like Dialog library. It is divided into user state, conversation state (used in group chats) and private conversation state. IBM Watson handles context through its visual Dialog Manager. It uses action variables and session variables. Action variables are short-term memory combined with actions created during slot filling. Session variables are long-term memory not tied to a specific action and persist throughout the user's interaction. Dialogflow in the CX version uses parameters for preserving context. These are separated between intent and form parameters. Both are related to slot filling, but intent parameters depend on intent and form parameters depend on a page form in Dialogflow terms. Rasa uses slot-filling and ML techniques to manage context and drive the conversation.

Slots are key-value pairs that are filled either through form filling or ad hoc by mapping slots to entities and intents or custom types during a conversation. Rasa uses policies to predict the following action. These policies consider context set through slots or using other means like ML techniques depending on the policy type. For example, Rasa offers TEDPolicy [20] (Transformer Embedding Debug Policy) and KerasPolicy, both context-aware ML models. Amazon Lex implements input and output context in combination with intents. An output context from one intent can be the input context for the next one. There are also session attributes which share the state between intents. Session attributes can only be defined through code. Finally, request attributes are used for sending data from the client app to the connected bot.

Entity extraction techniques: CDPs combine different approaches to match entities in users' utterances. Usually, they offer some built-in entities which cover common entity cases. DialogFlow supports recognizing entities with ML techniques or with pattern matching. The entities are categorized into built-in, custom and session entities. There is also a fuzzy matching feature, where NLU tries to identify an entity containing multiple words. Entities are mapped to parameters for slot filling. Amazon Lex does not distinguish between slot filling and entity extraction. There are built-in slots and custom ones in the same way as DialogFlow. Entity extraction with pattern matching is supported through a built-in type. There is also a grammar-type slot that is used in the case of speech recognition. Watson Assistant's latest version does not explicitly define entities. Instead, slot filling is applied through customer responses.

Table 2: cost of CDPs per request

System	text/request ($)	audio/request ($)
Dialogflow ES	0.002	0.026/min
Dialogflow CX	0.007	0.06/min
Amazon LEX	0.00075	0.026/min

Nevertheless, it uses an entity recognition mechanism under the hood to automate the process. It also supports built-in entities and pattern matching. Microsoft offers NER as a standalone service in Cognitive Service for Language. This service recognizes entities by using built-in types and ML per specific languages. Microsoft LUIS utilizes different entity types like regex entity, list entity, built-in entity and ML entities. Rasa offers other entity extraction methods for different use cases through components in the NLU pipeline, like EntitySynonymMapper for synonyms, whereas for dates, there is DucklingEntityExtractor. For pattern matching, RegexFeaturizer or RegexEntityExtractor can be used depending on the case. In RASA, the management of entity extraction is customized and can be fine-tuned by the developers of the chatbot

Price: The cost is a critical factor in evaluating a CDP. Rasa follows the open-source business model with an apache2 licence and charges extra features and technical support in its enterprise version. Dialogflow follows different plans for either CX or ES edition. ES edition offers a time-unlimited trial version, limited resources, and a paid plan. The paid plan charges a pay-as-you-go scheme for audio and text requests, plus for sentiment analysis and mega agent use. CX offers credits for a trial version and then charges for requests. IBM Watson Assistant offers a free Lite plan with limited resources. There are also the plus and enterprise-paid plans. The plus plan starts at 140$/month and depends on monthly active users—azure bot service charges with a pay-as-you-go scheme based on azure app services. For Cognitive Service, a free tier is offered with limits [21] (5.000 text records per month). Then the paid version depends on the number of text records added and the service used (language detection, custom question answering etc.). Amazon Lex has a relatively simple pricing scheme. There is a free limited plan for the first year. Then the paid plan is based on the number of requests. Dialogflow and Amazon Lex are based on requests, so basic direct price comparison is feasible as seen in Table 2

Sentiment Analysis: Depending on the situation, it may play an essential role during a conversation between a user and the chatbot. For example, if the user is nervous during a conversation, this is probably a sign for the chatbot to hand over the conversation to a human agent. All systems apart from Rasa offer sentiment analysis either as a standalone service or integrated through the NLU component. RASA does not offer a prebuilt part for sentiment analysis but provides the possibility to create a custom one. This component can either implement a custom-created model or call a standalone service created by other CDPs like Amazon Comprehend.

Benchmarking: A key factor to overall performance is correctly identifying intents and entities. The NLU component is responsible for recognizing intents and entities, and the benchmarks are conducted against it. The test should simulate actual life conditions and consider business requirements. According to H. Qi et al. [7], most cloud-based CDPs prohibit benchmarking their services, which may not be possible in the future. There are metrics from older attempts or comparisons of CDPs' NLU with pre-trained models. According to [7], Watson Assistant seems to have precedence over the other competitors (Amazon Lex was not included in the test). Finally, these benchmark tests were not performed against the latest versions of CDPs' NLU, like Dialogflow's CX version and Microsoft's Cognitive Services for Language. So, it is impossible to have a reliable conclusion based on those results.

6 FINAL RESULTS

First of all, during the course of work became evident that it is not possible to define the score of each CDP on criteria without taking into account the requirements of the chatbot project. Thus we should underline that the evaluation score as seen in Table 3 is based on requirements assumptions, and one could adjust the score based on the analysis of criteria implementation for each CDP described in the Descriptive Evaluation version.

We are proposing three new criteria, support of visual dialog, multiple agents and knowledge service. We noticed that the newer versions of CDPs emphasize the automation of dialog management by offering visual management of dialog flow and multiple agents architecture which further helps in the modularization of dialogs. There is also the tendency to introduce state-of-the-art models like BERT customized to real market challenges like training data limitations, non-standard user input robustness, and computational efficiency.

The next step is to use the AHP evaluation tool, as suggested by Pavel Kostelník and others [16]. This method adds weights to each criterion based on requirements exported by specific use cases as a decision-support tool. Overall, Microsoft and Dialogflow look like complete CDPs, even though total score differences between CDPs are relatively small. Also, Microsoft and Watson Assistant support every criterion up to some point. Dialog Flow fully implements most characteristics of any other CDP.

Table 3: Score summary of CDPs per criteria

	Visual dialog	Prebuild agents	Integration channels	knowledge engine	Language support	Development adaptability	Multi agents	Life cycle management	Testing	Debugging	Vendor locking	Run on-premises	STT & TTS	Context Management	NER	Price	Sentiment analysis	
LEX	2	4	3	3	2	3	3	3	2	2	3	1	3	3	3	3	4	47
Microsoft	4	2	4	3	4	4	4	4	3	3	3	2	3	4	4	2	4	57
Watson Assistant	4	4	4	4	3	3	3	3	3	3	3	2	3	4	3	2	3	54
DialogFlow	4	2	4	4	3	3	4	3	4	4	3	1	4	4	4	2	4	57
Rasa	1	1	3	2	3	3	3	3	3	4	4	4	2	4	4	4	2	50

REFERENCES

[1] E. Adamopoulou and L. Moussiades, 'Chatbots: History, technology, and applications', Machine Learning with Applications, vol. 2, p. 100006, Dec. 2020, doi: 10.1016/j.mlwa.2020.100006.

[2] A. Abdellatif, K. Badran, D. E. Costa, and E. Shihab, 'A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering', *IIEEE Trans. Software Eng.*, pp. 1–1, 2021, doi: 10.1109/TSE.2021.3078384.

[3] 'Webhook', *Wikipedia*. Apr. 23, 2022. Accessed: Jul. 19, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Webhook&oldid=1084306200

[4] 'IBM Cloud Docs'. https://cloud.ibm.com/docs/cloud.ibm.com/docs/watson-assistant (accessed Jul. 19, 2022).

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *arXiv:1810.04805 [cs]*, May 2019, Accessed: Dec. 05, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805

[6] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, 'Rasa: Open Source Language Understanding and Dialogue Management', *arXiv:1712.05181 [cs]*, Dec. 2017, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/1712.05181

[7] H. Qi *et al.*, 'Benchmarking Commercial Intent Detection Services with Practice-Driven Evaluations', *arXiv:2012.03929 [cs]*, Jun. 2021, Accessed: Mar. 07, 2022. [Online]. Available: http://arxiv.org/abs/2012.03929

[8] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, 'DIET: Lightweight Language Understanding for Dialogue Systems', *arXiv:2004.09936 [cs]*, May 2020, Accessed: Mar. 13, 2022. [Online]. Available: http://arxiv.org/abs/2004.09936

[9] D. Schnelle-Walka, S. Radomski, B. Milde, C. Biemann, and M. Mühlhäuser, 'NLU vs. Dialog Management: To Whom am I Speaking?', p. 4.

[10] A. Acharya *et al.*, 'Alexa Conversations: An Extensible Data-driven Approach for Building Task-oriented Dialogue Systems', *arXiv:2104.09088 [cs]*, Apr. 2021, Accessed: Mar. 28, 2022. [Online]. Available: http://arxiv.org/abs/2104.09088

[11] G. Arora, C. Jain, M. Chaturvedi, and K. Modi, 'HINT3: Raising the bar for Intent Detection in the Wild', in *Proceedings of the First Workshop on Insights from Negative Results in NLP*, Online, 2020, pp. 100–105. doi: 10.18653/v1/2020.insights-1.16.

[12] S. Larson *et al.*, 'An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction', *arXiv:1909.02027 [cs]*, Sep. 2019, Accessed: Apr. 27, 2022. [Online]. Available: http://arxiv.org/abs/1909.02027

[13] J. Sedoc, D. Ippolito, A. Kirubarajan, J. Thirani, L. Ungar, and C. Callison-Burch, 'ChatEval: A Tool for Chatbot Evaluation', p. 6.

[14] A. H. Miller *et al.*, 'ParlAI: A Dialog Research Software Platform', *arXiv:1705.06476 [cs]*, Mar. 2018, Accessed: Apr. 29, 2022. [Online]. Available: http://arxiv.org/abs/1705.06476

[15] S. Perez-Soler, S. Juarez-Puerta, E. Guerra, and J. de Lara, 'Choosing a Chatbot Development Tool', *IEEE Software*, vol. 38, no. 4, pp. 94–103, Jul. 2021, doi: 10.1109/MS.2020.3030198.

[16] P. Kostelník, I. Pisařovic, M. Muroň, F. Dařena, and D. Procházka, 'CHATBOTS FOR ENTERPRISES: OUTLOOK', p. 11.

[17] A. Patil, M. Karuppiah, N. A, and R. Niranchana, 'Comparative study of cloud platforms to develop a Chatbot', *International Journal of Engineering & Technology*, vol. 6, p. 57, Jun. 2017, doi: 10.14419/ijet.v6i3.7628.

[18] D. Braun and F. Matthes, Towards a Framework for Classifying Chatbots. 2019.

[19] A. Aghajanyan, X. Song, and S. Tiwary, 'Towards Language Agnostic Universal Representations', *arXiv:1809.08510 [cs, stat]*, Sep. 2018, Accessed: May 03, 2022. [Online]. Available: http://arxiv.org/abs/1809.08510

[20] V. Vlasov, J. E. M. Mosig, and A. Nichol, 'Dialogue Transformers', *arXiv:1910.00486 [cs]*, May 2020, Accessed: May 04, 2022. [Online]. Available: http://arxiv.org/abs/1910.00486

[21] 'Pricing - Language Service | Microsoft Azure'. https://azure.microsoft.com/en-us/pricing/details/cognitive-services/language-service/ (accessed Jun. 21, 2022).