CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

A Maturity Assessment Framework for Conversational AI Development Platforms

Master's thesis in Software Engineering and Technology

JOHAN ARONSSON

PHILIP LU

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2019

Master's thesis 2019

A Maturity Assessment Framework for Conversational AI Development Platforms

Analyzing conversational AI platform features and constructing a conversational maturity framework

JOHAN ARONSSON

PHILIP LU

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2019

A Maturity Assessment Framework for Conversational AI Development Platforms
JOHAN ARONSSON
PHILIP LU

Supervisor: Daniel Strüber & Thorsten Berger, Department of Computer Science and Engineering
Advisor: Alex Berman, Talkamatic AB
Examiner: Christian Berger, Department of Computer Science and Engineering

Master's Thesis 2019
Department of Computer Science and Engineering
Division of Software Engineering and Technology
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2019

A Maturity Assessment Framework for Conversational AI Development Platforms

JOHAN ARONSSON
PHILIP LU
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Conversational Artificial Intelligence (AI) systems have recently skyrocketed in popularity and are now used in many applications, from car assistants to customer support. The development of such systems is supported by a large variety of conversational AI platforms—all with similar goals, but different focus points and functionalities. Unfortunately, a systematic foundation for classifying conversational AI platforms is currently lacking. In this thesis, we propose a framework for assessing the maturity level of conversational AI platforms. Our framework is based on a systematic literature review, in which we extracted common and distinguishing concepts (called features) of various open-source and commercial in-house platforms. Inspired by language reference frameworks, we identify different maturity levels a conversational AI platform may exhibit in understanding and responding to user inputs. Our framework can guide organizations in selecting a conversational AI platform according to their needs, and platform developers in improving the maturity of their platforms.

Keywords: Model driven engineering, feature model, conversational AI, conversational maturity framework.

Acknowledgements

We would like to give our utmost thanks and sincere gratitude to our main supervisor Daniel Strüber who was always helpful and gave us guidance throughout the entire process of this thesis work. We would also like to extend our thanks to our co-supervisor Thorsten Berger for the support and guidance he gave us.

We would also like to thank Talkamatic and their entire team for giving us the chance to write our thesis with their company. Special thanks to our supervisor Alexander Berman who was very helpful and understanding.

In addition we would also like to thank our examiner Christian Berger for his comments and ideas on this thesis work.

Johan Aronsson & Philip Lu, Gothenburg, November 2019

Contents

List of Figures

List of Tables

1
Introduction

Conversational AI has recently surged in popularity and interest. A conversational AI system is an interface that can communicate and interact with users by relying on the automated processing questions and formulating answers. In 2016, Facebook announced a new platform to develop chatbots on their messaging application [1], which simplified the development of creating AI chatbots by providing relevant toolkits [2]. After that, many other companies have implemented chatbots for both text and speech. Three of the most popular conversational AI systems in existence today are Microsoft Cortana, Google Assistant, and Apple's Siri [3].

To support organizations in adopting conversational AI systems, a multitude of development platforms is available. By offering numerous concepts, such as natural language understanding, webhooks, and contexts, these platforms allow to engineer systems that can provide a rich, ideally almost human-like conversation experience. However, due to the large variety of available platforms, the relevance and need of each individual concept and its impact on the conversation experience are unclear. As a result, the use of such platforms may be overly complicated. To support organizations in selecting a suitable platform, and platform developers in increasing the maturity of their platforms, we need to improve our empirical understanding of the state-of-the-art of the domain. Specifically, we need to understand what platforms exist, what concepts they offer, what their concepts' characteristics are, in what combinations the concepts are used, and, in sum, what level of conversation they enable. Evaluating the conversational maturity that the different platforms offer might help in changing the perception that these systems are simply task-oriented tools, and that they can hold truly social conversations. Additionally, it may help in understanding how the more functional terms of these platforms relate

to the conversational ability [4].

In this thesis, we provide a maturity assessment framework for conversational AI platforms. We provide a comprehensive overview of the features available in today's platforms and analyze these features to see how they relate to the quality and ability of conversational AI systems produced using them. Finally, inspired by human language development frameworks, we propose a layered framework with multiple levels of conversational maturity. With this contribution, we aim to improve our empirical understanding of current development platforms for creating conversational AI systems, their concepts, and the level of conversation that bots created with these systems can achieve. As a benchmark for assessing existing and new platforms, our framework can support and guide practitioners who engineer such platforms. Moreover, it can help researchers understand the concepts that exist, identify gaps between practice and research, and scope future research. In the long term, this could help in creating better conversational AI systems.

We address the following research questions:
RQ1: *What platforms exist for engineering conversational AI systems?*
RQ2: *What are the features of these platforms?*

These first two questions are aimed towards analyzing existing conversational AI platforms and extracting information regarding their usage and features. A specific focus is on the platforms' ability to model conversation dialogs.

To this end, we performed a literature study, in which we collected papers presenting different platforms. We then analyzed the documentation of the platforms to identify their distinguishing characteristics and concepts (features). To provide an intuitive, hierarchical overview on the multitude of available features, we grouped them into a feature model [5, 6], a common notation for modeling the variability of portfolios of software systems [7] in a domain. Feature models have also become popular in empirical studies for modeling the design

space of technologies, such as model transformations [8] or language-engineering workbenches [9].

RQ3: *How to systematically evaluate the conversational maturity level of a conversational AI development platform?*

We created a framework that can be used to evaluate the conversational maturity (intuitively, how "smart" an agent is in understanding questions and formulating responses) offered by the platforms. To this end, we considered existing frameworks that evaluate the language proficiency of humans, and previous discussions on how to evaluate different conversational AI platforms. We then devised a framework based on the features identified in the first research questions and their effect on the human-like performance of a conversational AI platform.

There are not many studies on the conversational maturity that different conversational AI platforms offer. One of the few exceptions is the one by Venkatesh et al. [10], who describes how to evaluate the conversational maturity of conversational agents in terms of certain metrics. In contrast, our work focuses on recently available platforms and on how their features impact the conversational maturity of systems created upon these platforms (cf. Section 2 for a discussion of related work).

2
Background

2.1 Overview

With the recent developments in many of the fields that conversational AI is built upon, including dialog management and natural language processing, many different conversational AI systems have emerged [11]. Within industry, this technology has been incorporated into search engines, mobile devices, and personal computers. In search engines such as Google and Bing, conversational AI is used to create the feeling of having a conversation with the search engine, enhancing the experience. In mobile devices and personal computers, one use of conversational AI is to create virtual assistants. Some of the biggest virtual assistants on the market today are Apple's Siri, Google Assistant, Amazon Alexa and Microsoft Cortana [12]. These assistants also have the capability of acting as chatbots where they keep a turn-based dialog (a dialog where the user and the bot take turns in asking and responding to queries) with the user. There also exist conversational interfaces that only focus on this type of dialog-based conversation such as XiaoIce [13] and Replika [13]. These dialogs use what is known within conversational AI as *intents* and *entities* to understand the user's goal behind the query. In other words, an *intent* is what the user wants to achieve with the query, and an *entity* is the key information for answering the intent.

Recently a number of different platforms have been made available to simplify the creation and integration of conversational interfaces for developers. The most popular ones are: Google's DialogFlow (formerly api.ai)[1], IBM's cloud-based bot service Watson Conversation[2],

[1]https://dialogflow.com/
[2]https://www.ibm.com/cloud/watson-assistant/

Amazon Lex[3] and the Microsoft Bot Framework[4]. These platforms come equipped with several different technologies used for Natural Language Understanding, Dialog Management, response generation and other aspects [13, 14]. Figure 2.1 show an overview of how these technologies can be connected in a conversational AI system.

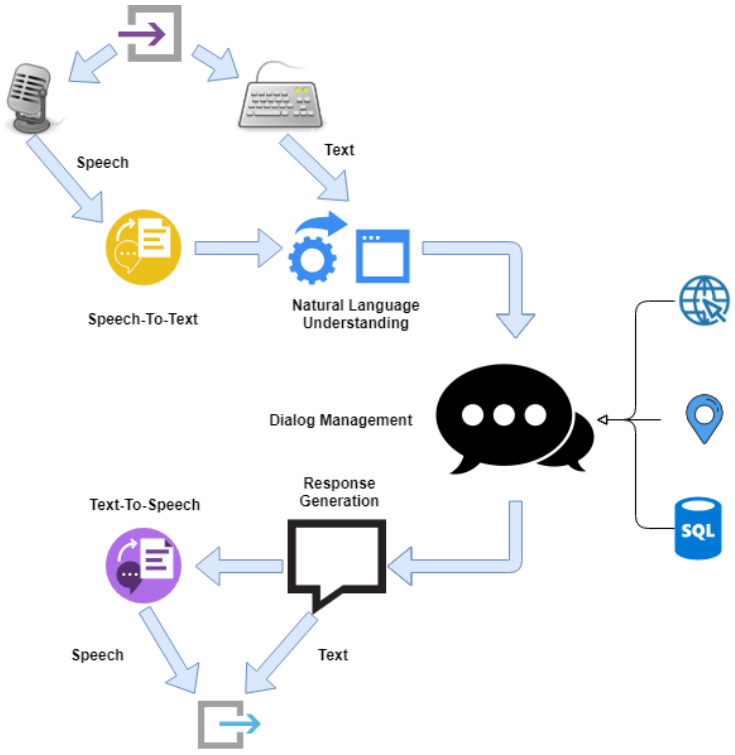

Figure 2.1: Overview of the flow of conversational AI systems

As can be seen, the flow starts with an input, in most cases, this will be either as text or speech supplied by the user. For the natural language understanding to be able to process a supplied speech input it first has to be processed into a format that can be understood. This is done using automatic speech recognition technology that will output a text given an input of audio. Recent advances in the accuracy of speech recognition [15] have been a big reason for the recent emergence and popularity increase of conversational AI systems and platforms. These advances allow for an overall quality increase in the conversational abilities of conversational systems.

The input will be processed by a natural language understanding ser-

[3]https://aws.amazon.com/lex/
[4]https://dev.botframework.com/

vice that will map out and find certain keywords that will help the conversational AI system understand what a user is asking for i.e. *intents* and *entities* mentioned earlier [16]. Examples of how these are defined and exactly what they are will be described more in section 4.

When the objective of the user has been determined the conversational system needs to know what to do to fulfill the user request. This is done through dialog management as seen in figure 2.1. Depending on what the user is asking for the AI might need to collect information from either a website or a database. In more complex systems it might be necessary for the AI to recollect information from a previous conversation and integrate it to the response of the current user query [17]. It is also the job of dialog management to determine the timing of the response to the user to create a seamless conversation. Each system may behave differently depending on who is to use it and where it is to be implemented. As such the dialog management is where most manual development work is done when creating conversational systems on platforms such as those mentioned above [18].

Once the proper response has been constructed it needs to be made into something that can be understood by a user. This is done through response generation which can be done using two different methods. The first involves using predefined answers which takes the information gathered through dialog management and assigns it to certain slots of the response. Another method is to use natural language generation that use deep learning, more specifically recurring neural networks, to build responses using human-like language [19]. This is generally used when more complex information needs to be presented in a way that is understandable by the user. In a query where many different options can be chosen i.e. when asking for the time-table of a bus, predefined answers may not be clear enough or require much effort for developers to define. Natural language generation allows for creating responses that give some information at a time or filters information so that only what is needed, is presented to the user. Depending on the system this output can be either in text or speech, it is also possible to respond with images and other media [18].

2.2 Related work

Patil et al. [20] makes a general comparison of features and functionalities between some of the platforms mentioned above, giving an overview of what platform one might choose for developing a conversational AI system. There have also been more specific studies conducted which compare the natural language understanding and conversational abilities of these types of platforms. In a study by Massimo Canonico and Luigi De Russis [21], they compare the natural language understanding performance these platforms have in terms of usability, pre-built intents (a number intents already existing in the natural language understanding tool), context etc. Braun et al. [16] evaluates the natural language understanding capabilities of systems such as LUIS (natural language understanding service used with Amazon LEX) and Watson Conversation by training it on sets of data and compared their results to a set standard.

McTear [22] describe the two main conversation models "one-shot queries" and "slot-filling dialogues". He compares different platforms ability to handle follow up questions in one-shot query scenarios and their mechanisms for slot-filling (a type of conversation where the bot asks specific questions to fill certain slots to fulfil a user intent). McTear also presents a number of problems that developers may face when creating conversational interfaces with these platforms. One of the main issues is that it might be difficult to know what functionalities a specific platform offers. There is also a difficulty in interpreting what functionalities might be common between platforms since there is no standard terminology.

Venkatesh et al. [10] describe a number of metrics that can be used to evaluate the overall performance of a conversational agent based on the annual competition Alexa Prize [23] made for furthering conversational AI. They propose metrics such as conversational user experience, engagement, and conversational depth to measure the conversational abilities of entire conversational AI systems or chatbots [10]. Shawar and Atwell [24] describe metrics to specifically evaluate chatbot systems, a type of conversational AI interface. They argue that

metrics for evaluating the abilities of these systems should be done based on the application and its domain and not solely on a standard.

One of the main issues with creating the metrics described above is the understanding of what a good conversation is. Clark et al. [4] discuss that people generally describe conversations with conversational interfaces in terms of their performance and perceive them more as a device to be controlled. Indicating that people have a previous notion of how these systems will behave coming from a perception that infrastructure to support proper human-to-human dialogs do not exist.

3
Methods

In this chapter the methods used for this research will be presented. These methods will also be explained as to why it was used.

3.1 Identification of Conversational AI development platforms

In the first part of our study (RQ1), we aimed to explore the variety of existing conversational AI development platforms. To this end, we used several methods as follows: Literature review, Information sourcing, Database search.

3.1.1 Systematic Literature Review

We used a systematic literature review to identify papers on conversational AI systems. We focused on papers that present and evaluate platforms used to develop such systems. Specifically, among the different methods that exist for conducting literature reviews, we used *snowballing*. We followed Wohlin [25], who describes two types of snowballing and provides guidelines for performing them: forward and backwards snowballing. We performed backwards snowballing, to find papers describing conversational AI platforms. Backwards snowballing involves selecting a number of papers to be used as a start set to find more relevant papers in the same field by tracing the reference lists of the papers. The start set should include a number of different papers from different areas of the field, different authors, and different points in time. The idea is to cover the considered field or topic to the largest possible extent. The reference lists of the papers in the start set are then evaluated based on certain inclusion and exclusion criteria (explained shortly). From the start set, additional papers can be found, which we also screened. Each set of reviewed papers is one

iteration of the snowballing procedure, once no more papers can be found the process is over [25].

We collected the start set for snowballing through database searches, using search strings such as "Conversational AI," "Conversational AI development platforms," and "Chatbot platforms". The full list can be found in table A.1 in appendix A. The first 50 search results of each of these search strings were examined based on the criteria listed below. To determine whether to include or exclude a certain paper we used the following procedure: Read the title and abstract and skim the whole paper to determine if any conversational platforms can be found. A paper could be excluded at any stage of the process based on its relevance to the study. The databases used in this search were Google Scholar [1], IEEE [2], arXiv [3], Springer [4] and our university's library.

Our inclusion criteria were:

- Papers published after 2000, after which most recent platforms have been developed, were candidates for inclusion.
- Papers examining and presenting different conversational AI/bot platforms were included.
- Papers that only examine characteristics of conversational AI and do not mention any platforms were excluded.

The platforms that were found through the literature review were then examined in order to determine if enough information about them was available to fairly assess what features the platforms provided.

To this end, our exclusion criteria were:

- Platforms that are no longer available or heavily outdated were excluded.
- Systems that were either simple chatbots or just single agents were excluded.
- Platforms that did not have enough documentation available pub-

[1] https://scholar.google.se/
[2] https://ieeexplore.ieee.org/Xplore/home.jsp
[3] https://arxiv.org/
[4] https://link.springer.com/

licly were excluded.
- Platforms that did not have a strong enough user base, either by individuals or companies or both, were excluded.

3.1.2 Information Sources

In addition to the snowballing procedure, we consulted an industrial partner—a company with years of experience in conversational AI to find platforms that we might have missed. The practitioners gave valuable information regarding existing platforms the company was aware of, and regarding these platforms' specifications. Our partner also provided additional detailed information about platforms the company identified before, together with the platforms' characteristics.

3.1.3 Database Searches

To find platforms outside the more formal channels of published literature and company expertise, we also sought via the Google search engine. This required specific care and source criticism, since the information available may be outdated or even false. We conducted the searches using search terms that try to find platforms similar to those found through previous methods, for instance, "DialogFlow competitors". A list of all search terms used can be found in table A.2 in appendix A.

3.2 Documentation Analysis

The main process for collecting information about the different conversational AI platforms was document analysis. Document analysis involves going through any documentation available for a specific entity, such as a software platform. It allows for the collection of data that later can be evaluated and grouped based on certain criteria. Document analysis is often quite efficient and cost-effective since no new data needs to be acquired; instead, already existing data is evaluated. However, there is a risk that the documentation may be incomplete [26].

We analyzed the documentation available for all considered platforms to identify their common and distinguishing features, thus addressing RQ2. Whilst the entire platforms were analyzed to be able to give an overview of the entire system structure, we put special emphasis on their *conversation-defining features*. The conversation-defining features build up the dialog management portion of the platforms, which define what the bot can understand and how it should respond. This process also helped us mapping similar features whose names vary between different platforms.

3.3 Feature Model

To represent the identified features, we developed a feature model [5]. Feature models visualize the features of a platform by displaying them in a hierarchy, thus providing a good overview of top-level and more fine-grained features. Features are shown as mandatory or optional. Mandatory features are largely common in all different systems, while optional features exist in just one or a few of them. The model also includes constraints between the features, such as *dependencies*, in which a feature needs another feature for its implementation. There are a few other models that can be used for similar purposes, such as class diagrams. However, we used feature models since they provide a compact, hierarchical overview, which is good for managing complexity in large systems [27], and since feature models have been used in similar empirical studies [8, 9]. Finally, to create the feature model, we used FeatureIDE[5], one of the standard tools in this area.

3.4 Designing the Conversational AI Maturity Framework

3.4.1 Identification of Language Maturity Frameworks

As a prerequisite for creating a maturity framework for conversational AI development platforms, we explored if any similar attempts had been made before. We performed a literature review to identify any existing frameworks, either directly related to conversational AI classification or to evaluate the conversational maturity of a human. We

[5]http://www.featureide.com/

searched using Google Scholar, IEEE, arXiv, Springer and our university's library. The following search phrases were used when looking for these frameworks: "Common language framework", "Human language framework", and "Language framework". The full list of phrases can be found in appendix A. From these searches, the top 50 results were considered to determine their relevance for this study. We used the following exclusion and inclusion criteria:

- Papers discussing different aspects of what makes good conversational AI were included.
- Papers with frameworks used to evaluate maturity of either human or bot conversation maturity were included.
- Papers that have metrics for evaluating conversational AI systems were included.

To determine whether a paper matched the criteria above the following procedure was followed: read the title of the papers; read the abstract of the papers; read discussion to determine if any frameworks are presented or characteristics of good conversational AI are mentioned. As mentioned above a paper could be excluded at any point of the process, if the title was out of scope the paper is directly excluded.

3.4.2 Designing the Framework

Our goal was to create a framework that describes various levels of maturity, all building upon each other. To identify the maturity levels, we used the features found through the documentation analysis. We decided for each feature if it contributes to conversational maturity, and clustered those features that do into distinct, progressive levels. We took further inspiration from three language expertise frameworks: Common European Framework of Reference (CEFR, [28]), American Council on the Teaching of Foreign Languages (ACTFL, [29]), proficiency guidelines and the Interagency Language Roundtable (ILR, [30]). Furthermore, we considered additional papers that compare and evaluate the different language frameworks, to get an overview of how they perform and differentiate [31, 32, 33].

4
Results

We present the results from our literature review, the documentation analysis performed on the identified conversational AI platforms, and the obtained feature model with common and distinguishing features of these platforms. Lastly, we will present the final conversational language maturity framework for conversational AI platforms.

4.1 Result of Literature Review

The database searches resulted in 10 sources which were used as the start set for the snowballing procedure. The references of each of these papers were then screened in order to find any other papers relevant for the purpose of finding conversational AI platforms. The snowballing was ended after three iterations of this procedure.

In the first iteration, based on the start set, 13 additional papers were added. The list of potential candidates were narrowed down by using the following procedure: Read the title of papers; check where the paper is referenced in the text; read abstract of papers; look at the full text to determine if it contains any new conversational platforms. The place of reference was checked in order to determine if it was used in conjunction with text that describes conversational platforms. All papers were matched against the same criteria that were used to put together the start set, see Section 3. The second iteration of the snowballing procedure was done on the 13 newly found papers. From these papers, another 3 were identified that describe conversational AI platforms. The third and last iteration identified no additional relevant papers. A list of all the identified papers can be found in appendix B.

Using snowballing, we identified a total of 56 different potential conversational AI platforms. From these 56 platforms, we removed a

number of duplicates arising from the same system appearing under different names: API.ai was renamed to DialogFlow, and IBM voice server and AT&T watson were the predecessors to IBM Watson Conversation. We excluded the conversational interfaces Cortana, Google assistant, and Amazon Alexa, as they are not actual development platforms. Cortana is developed by Microsoft who makes its technology available through Microsoft Bot Framework. Google assistant is very much related to DialogFlow and Amazon Alexa with Amazon Lex. The remaining platforms were matched against the inclusion and exclusion criteria mentioned in Section 3. These criteria were used to narrow down the set to a total of 10 platforms: DialogFlow, Microsoft Bot Framework, Houndify, RASA, Amazon Lex, IBM Watson Conversation, VoiceXML, Recast.ai, Kore.ai, and AIML.

After the snowballing had been performed an individual at the partnering company was consulted to find any platforms that might have been missed during the snowballing. It resulted in the addition of three new platforms that had not yet been examined as well as the confirmation that the platforms found in the literature review match many of the platforms that had been found by the company. The three new platforms that were found are: Teneo, Boost.ai, and TDM. Teneo and Boost.ai were removed from further investigation as they lacked sufficient documentation of their contained features.

The last avenue to identify conversational AI platforms were through searches in the Google search engine. The searches made here identified a number of new platforms that had emerged quite recently. Among the searches only three qualified as the type of conversational AI platform that is evaluated in this thesis: Meya, Chatbot and Botpress. Chatbot and Botpress lacked available documentation supporting a fair assessment of the functionality available within the respective platforms. For this reason, both of these platforms were excluded. Meya was included since its documentation was extensive enough to form a full image of its features.

We finally obtained a list of twelve systems to further analyze. These, together with some core characteristics, can be seen in table 4.1.

Table 4.1: Identified Conversational AI platforms

Platform	Open/closed source	Availability	Modality
DialogFlow	Closed	Semi-commercial	Web-based
Meya.ai	Closed	Commercial	Web-based
Microsoft Bot Framework	Closed	Semi-commercial	Web-based
Houndify	Closed	Semi-commercial	Web-based
Amazon Lex	Closed	Commercial	Web-based
RASA	Open	Free	Command-line
IBM Watson Conversation	Closed	Semi-commercial	Web-based
VoiceXML	Open	Free	Implementation dependent
Recast.ai	Closed	Semi-commercial	Web-based
Kore.ai	Closed	Semi-commercial	Web-based
AIML	Closed	Free	Implementation dependent
TDM	Closed	Commercial	Command-line

4.1.1 Results of the Documentation Analysis

We thoroughly analysed the documentation of platform to pinpoint its included functionality, resulting in a list of features. These features were then reviewed to identify common and distinguishing features between the different platforms. If the same feature was found in multiple platforms under different names, we continued with the name used more often in the platforms and existing literature. The consolidated features were added to a feature matrix which is described in 4.1.1.1. The end result was a list of 54 different features that we grouped and organized in a hierarchy to obtain a feature model, discussed in what follows.

4.1.1.1 Feature matrix

After the documentation analysis, we then composed all of the features into a feature matrix to get a good overview of which platforms supported which features. This also gives a good understanding of the different complexities of the platforms. The complete feature matrix

can be seen in table 4.2 below. This table show the feature name and the platforms analyzed, if a platform supports a specific feature it is marked with a circle.

Table 4.2: Feature matrix

Feature name	RASA	SAP	Watson conversation	LEX	Houndify	DialogFlow	MBF	Meya.ai	TDM	VoiceXML	Kore.ai	AIML
ContextualDialogs	●	●	●	●	●	●	●	●	●	●	●	●
WebIntegration	●	●	●	●	●	●	●	●	●	●	●	●
DialogDefinition	●	–	–	–	●	–	–	–	–	–	–	–
SlotFilling	●	●	●	●	●	●	●	●	●	●	●	●
AutomaticUnderstanding	●	–	●	●	–	●	●	–	●	–	●	–
ErrorFeedback	●	●	●	–	–	–	–	–	●	–	–	–
VisualisationTools	●	●	●	●	–	–	●	●	●	–	●	–
FallbackActions	●	●	●	●	●	●	●	●	●	●	●	●
EntityDefinition	●	●	●	●	–	●	●	●	●	●	●	●
IntentDefinition	●	●	●	●	●	●	●	●	●	●	●	●
Affirmation	●	●	●	●	●	●	●	●	●	●	●	●
Rephrasing	●	●	●	●	●	●	●	●	●	●	●	●
DebugTool	●	–	●	–	–	–	●	–	–	–	–	–
ModelEvaluation	●	–	–	–	–	–	–	–	–	–	–	–
Policies	●	–	–	–	–	–	–	–	–	–	–	–
CustomTrainingData	●	●	–	●	–	●	–	–	–	–	–	–
TrainingData	●	●	●	●	–	●	●	●	●	●	–	–
Synonyms	●	●	●	●	–	●	●	–	●	●	●	●
OpenQuestion	●	●	●	●	●	●	●	–	●	●	●	●
SpeechInput	●	●	●	●	●	●	●	–	●	●	●	–
TextInput	●	●	●	●	●	●	●	●	●	●	●	●
ImageInput	–	●	–	●	–	–	–	●	–	●	–	–
URLInput	–	–	–	–	–	–	–	–	–	●	–	–
SocialPlatformSupport	●	●	●	●	–	●	●	●	●	●	●	–
FrontendIntegration	●	●	●	●	●	●	●	●	●	●	●	●
VoiceActivityDetection	●	●	●	●	●	–	–	–	–	●	●	–
MultiProgrammingLanguage	–	●	–	●	–	●	●	–	–	–	–	–
MultiLanguage	–	●	–	●	–	●	●	●	●	●	●	–
Sentiments	–	●	●	–	–	–	●	–	–	–	–	–
ImageOutput	●	●	●	–	–	●	●	–	–	–	–	–
TextOutput	●	●	●	●	●	●	●	●	●	●	●	●
SpeechOutput	–	●	●	●	–	●	–	–	●	●	●	–
ListOutput	●	●	●	●	●	●	–	–	●	●	–	–
SystemVersioning	–	–	●	–	–	●	–	–	–	–	–	–
TopicShifting	–	–	●	–	–	–	●	●	–	●	●	–
PredefinedSlotTypes	–	–	●	–	–	–	●	●	–	–	–	–

Table 4.2: Feature matrix

Feature name	RASA	SAP	LEX	Watson conversation	Houndify	DialogFlow	MBF	Meya.ai	TDM	VoiceXML	Kore.ai	AIML
ToneAnalyzer	–	–	●	●	–	–	–	–	–	–	–	–
Translation	–	–	–	●	–	–	●	–	–	–	–	–
ContentCatalogs	–	–	–	●	●	●	●	–	●	–	–	–
SpellingCorrection	–	–	●	●	–	●	●	–	–	●	●	●
MemoryForContext	●	●	●	●	●	●	●	●	–	●	–	–
DialogInitiation	●	●	●	●	●	●	●	–	●	●	●	●
MultipleUserIntents	–	–	●	●	–	●	–	–	–	●	●	–
LanguageSeparation	–	–	●	–	–	●	●	–	●	–	–	–
MultipleConversationDomains	●	●	●	●	●	●	●	–	●	●	●	●
SearchorientedDialog	●	–	●	●	–	●	●	–	●	●	–	–
Propositionality	–	–	–	–	–	–	–	–	●	–	–	–
LanguageRecognition	–	–	●	●	–	–	●	–	–	–	–	–
YesNoQuestion	●	●	●	●	●	●	●	●	●	●	●	●
DialogSpecification	●	–	●	–	●	–	–	–	–	–	–	–
NrOfEntities	–	●	●	–	–	●	●	●	–	–	–	–
NrOfIntents	–	●	●	–	–	●	●	●	–	–	–	–
TrainingPhrases	●	●	●	●	–	●	●	●	●	●	●	●
OneShotQueries	●	●	●	●	●	●	●	●	●	●	●	●
SmallTalk	●	–	●	●	–	●	●	–	●	●	–	–

4.1.2 Feature Model

Figure 4.1 shows a high-level overview of our feature model, highlighting its four top-level feature groups: *System*, *Conversation*, *Input modalities*, *Output modalities*. These top-level feature groups and their contained features will be discussed in this section.

We use the standard syntax of feature models. Specifically, as shown in the legend, features can be marked as mandatory, meaning that they exist in all or most of the analyzed platforms, or optional, meaning that they only exist in some platforms. Numbers attached to a node indicate that the node, in fact, represents a collapsed sub-tree, with the specified number of total nodes in the sub-tree. Abstract features are used for grouping purposes. "Or" features are used to specify features groups, where each considered platform had at least one of several features of the group. Descriptions of all features can be found in appendix C.1. In what follows, we describe the most crucial features, including those deemed as particularly relevant for assessing conversational maturity.

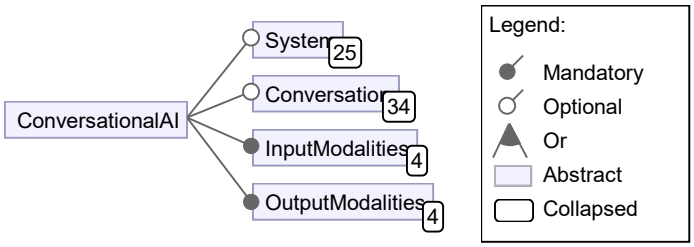

Figure 4.1: Top level view of the feature model

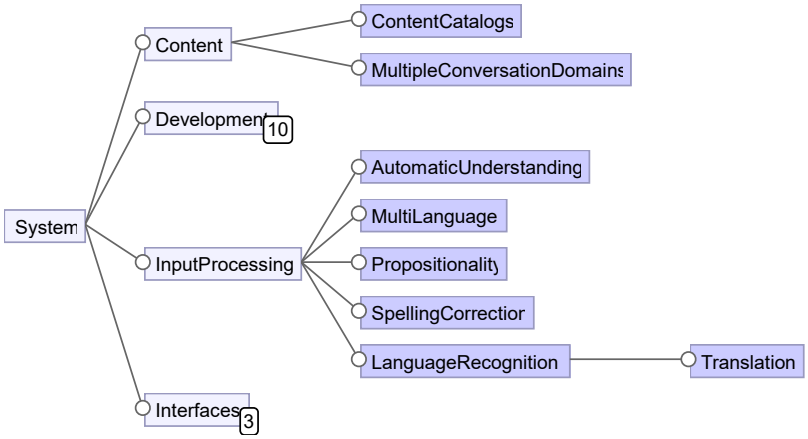

Figure 4.2: Main system features

4.1.2.1 System Features

From multiple *System* features as shown in Figure 4.2, *Content* refers to the different conversation contents and how the platform handles them. A conversation content can be, for example, “phoning a friend” or “weather information”. Figure 4.3 shows two crucial sub-features of content: *ContentCatalogs* refers to platforms with in-built content catalogues to simplify the development of the conversational AI bot. These catalogs contain entities and intents that are common in the domain.

MultipleConversationDomains is used to distinguish platforms that support the handling of domains that are completely independent from one another, thus making it possible to have 2 different content domains in the same conversation. This is in contrast to most platforms, which only support one particular domain with no explicit separation from other domains.

MultiLanguage is the conversational AI feature that regards to the number of supported languages within the platform.
Development features concern the development process of systems using the platform, which can be supported by features such as error feedback, debugging, and versioning tools.

Input processing refers to features to processing of the user input, such as *SpellingCorrection*. Different *interfaces* being supported may include a custom frontend, integration with social media, and other websites.

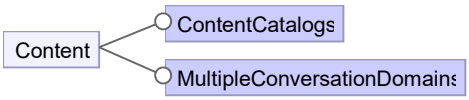

Figure 4.3: Content features

4.1.2.2 Conversation Features

One of the main reasons why there are so many different conversational AI platforms is that most handle conversations differently from one another. These differences can be anything from the content of the

conversation to the dialog management of the platform. The platforms consider conversations differently depending on what the intent of use is and the area of use is. Many of the different platforms in the market are focused on one specific field of expertise and are customized to fit the needs and standards of this field. An overview of these features and feature categories can be seen in Fig. 4.4. These features are all regarding the conversation between the agent and a user, everything from processing to supported conversation types. Language-specific features are also in the *Conversation* category, since the language is a part of the conversation.

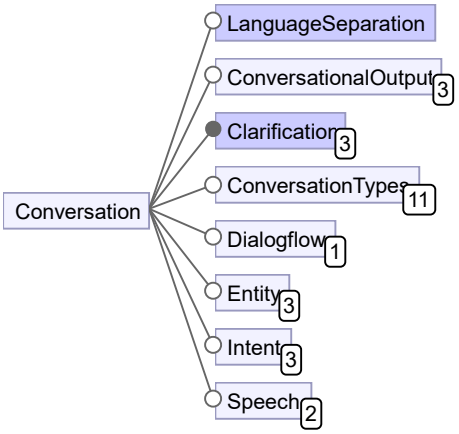

Figure 4.4: Conversational features

LanguageSeparation is the ability of the system to separate language-specific and non-language-specific information from a sentence. This allows the system to identify what parts of the sentence is crucial for the information to be received and what parts are not. This feature will simplify the translation of a conversation and the multilingual maintenance of the system.

Conversational output features, depicted in Fig. 4.5, affect how the conversational output is processed. *DialogInitiation* is one way to do so, it allows the developer to instigate a conversation. Different companies have different *Policies* and rules to adhere so some platform allows for such policies and rules to be implemented within the system itself. *Sentiments* allows for the conversational AI to display emotions in their response.

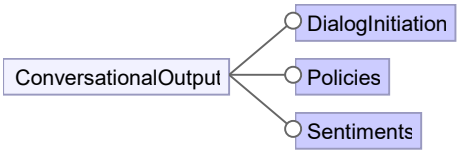

Figure 4.5: Conversational output features

Clarification is something that all the analyzed platforms support in some way, since not misunderstanding the user is crucial for supporting the robustness of the system. As summarized in Fig. 4.6, this is done by using *Affirmation*, *Rephrasing* or *FallbackActions* to confirm the users intent. These features affect how the system reacts when a user input is not understood or if a user input can be assigned to two different intents. These features also allow giving the user a second chance to change their mind or query.

Figure 4.6: Clarification features

ConversationTypes features cover the different types of conversation and questions that a platform supports. These features and feature subsets can be seen in Figure 4.7.

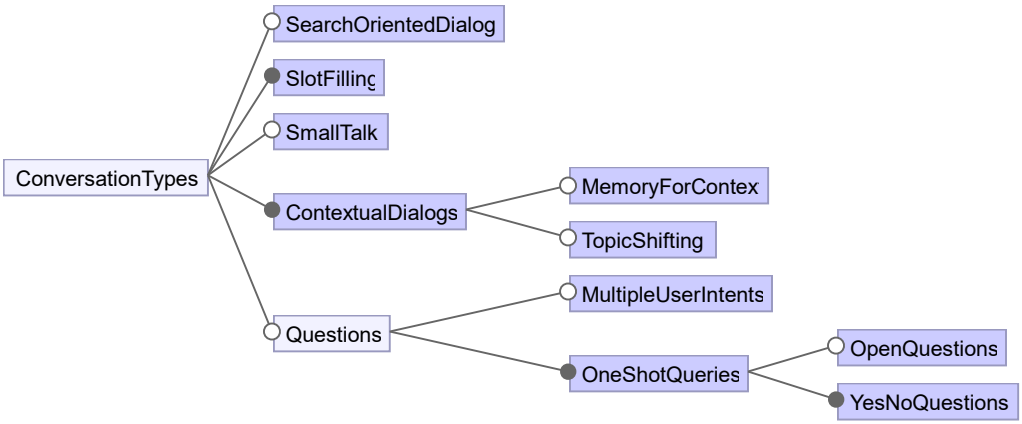

Figure 4.7: Conversation types

SearchOrientedDialog refers to a dialog that searches through a database to find matching entities and respond to a user intent.
SlotFilling is a dialog where to conversational AI ask for additional information to fill certain criteria to match the correct intent to an entity. An example of this could be:
User: What is the weather like today?

Bot: Which city would you like to search the weather for?
User: New York.
Bot: The weather in New York is cloudy.

SmallTalk is a conversation type that refers to conversations without any specific end goal from the user. These types of conversations can be anything from asking how you feel to telling jokes.

ContextualDialogs are an important component of conversational AI systems, supported by every analyzed platform. We found two different ways to support contextual dialogs: one or multiple contexts per conversation. To support contextual dialogs a conversational AI platform must have the feature *MemoryForContext*. MemoryForContext is specially allocated memory that the AI uses in order to remember previous information. Multiple context is referred to as *TopicShifting* and can enable conversations like, for example:

User: Send a text message to Peter.
Bot: What would you like to text?
User: I want to book a flight to Japan for tomorrow.
Bot: What time would you like to book the flight at tomorrow?
User: At 4am.
Bot: Where in Japan would you like to fly?
User: I would like to text: "The temperature in New York is 20°C."
Bot: Ok, your message has been sent.
User: I would like to fly to Tokyo.
Bot: Ok, flight booked to Tokyo tomorrow at 4am.

The last conversation type is *Questions*, these can vary from simple *OneShotQueries* to complex *MultipleUserIntents*. There are two types of OneShotQueries: *YesNoQuestions* and *OpenQuestions*. Both these types only require one response from the conversational AI to fully answer the query. MultipleUserIntents are queries with multiple intents within them, an example can be: "What is the time and weather like in New York?"

Features related to *Intent* are concerned with intent manipulation and intent restrictions. Intents are used to define the users' goal with the query, for example:

User: What is the time in New York?

Bot: The time in New York is 1pm.

In this case the intent of the user is: finding out the time. These features can be seen in Figure 4.8.

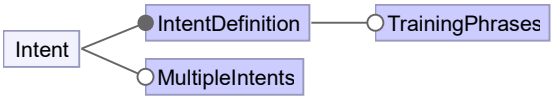

Figure 4.8: Intent features

Entity features, as shown in Figure 4.9, are concerned with entity manipulation and entity restrictions. Entities are descriptive actions the conversational AI can perform after identifying the users intent. An example is:

User: Can you call mom?

Bot: Calling mom.

In this case, the intent would be to make a call and the entity would be mom. Entity and intent are two of the features that is underlying to most functions within conversational AI, since it allows for the conversational AI platform to understand the user [16].

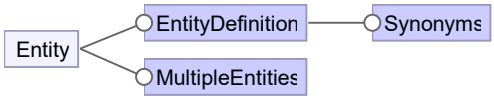

Figure 4.9: Entity features

Several platforms support features specific to *Speech*, one such feature is *VoiceActivityDetection* which allows the system to detect changes in audio level to determine whether or not the user is currently speaking. Another feature is *ToneAnalyzer* which allows the conversational AI to identify emotions within the speech pattern. These features, depicted in Fig. 4.10, are required by those platforms that support speech as an input modality.

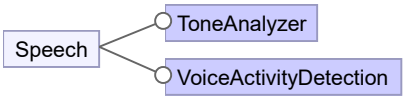

Figure 4.10: Speech features

4.1.2.3 Input Modalities

For a user to communicate with a conversational system the platform must support reading and analyzing one or several input types. The different input modalities supported by the considered platforms are shown in Figure 4.11: input of text, speech, images, and URLs. The larger the number of input modalities a platform supports, the larger the potential areas of use and accessibility of the created systems.

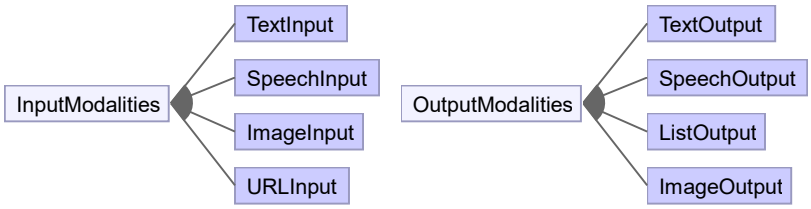

Figure 4.11: Input and output modalities

4.1.2.4 Output Modalities

For a two-sided conversation, the conversational AI system must be able to answer to respond to user queries by using one or several output modalities, the output modalities can also be seen in figure 4.11. These different types of output, like input modalities, allow the system to be used in different types of environments. Some environments only allow for one type of output to not disrupt the user's focus. For example, in a moving car, the optimal type of output would be speech. To have several types of output to choose from will both enhance the usability and the accessibility of the developed system.

4.2 Maturity Assessment Framework

To support the evaluation of the conversational maturity of a conversational AI platform, we created a maturity framework. The framework is inspired by frameworks used for human language development such as CEFR, ILR and ACTFL, which help to assess the conversational maturity of a language learner. These different language frameworks are all commonly used as guidelines, to assess the language development within humans. They are all based on the principle of using subsequent levels to evaluate language development. Some frameworks have more levels than the others but the base principle is the same

within all of these. We also followed this structure for the conversational maturity framework for conversational AI platforms, to be evaluated for a specific level all the preceding levels must be encompassed. This maturity framework can be used in similar fashion to the human language frameworks, but with the focus on conversational AI platforms, to evaluate the conversational maturity level of a conversational AI platform and how it compares to other platforms.

We obtained the framework by considering all features and terminologies of the state-of-the-art systems surveyed in the study, which can be seen in table 4.1. For each feature and terminology, we checked if it is relevant to conversational maturity. We then grouped the features deemed as relevant to different maturity levels, each level representing a different stage of conversational maturity. The resulting framework, as summarized in table 4.3, is divided into four main levels. Each of these subsequent levels are then divided into two sub-levels for further differentiation. The lower levels indicate a low conversational maturity with very simple response patterns, comparable to early stages of language development in humans. Higher levels signify more complex understanding and response capabilities, which can be compared to later stages of development where humans already are proficient with the language. For a conversational AI platform to be assigned as a specific level it has to accommodate all the preceding levels of conversational maturity. So the framework builds upon the preceding levels and has each of these levels as a prerequisite for the next. For example: to be assessed as level 2a, the platform must satisfy all criteria for levels 1a and 1b, in addition to those of 2a. The levels are proposed as followed:

Level 1 Indicates very limited conversational abilities and very simple comprehension.

- **Level 1a** Capabilities within the explicitly defined domain and response restricted to isolated words.
- **Level 1b** Capabilities within related domains and response restricted to sentence fragments.

Level 2 Indicates abilities to hold a short conversation with context and comprehension on a intermediate level.

- **Level 2a** Capabilities to understand longer queries and respond-

ing with full sentences.

Level 2b Capabilities to memorize context for a short conversation and can respond with questions.

Level 3 Indicates abilities to hold a long conversation with context and comprehension on a advanced level.

Level 3a Capabilities to comprehend spelling mistakes and small talk.

Level 3b Capabilities to comprehend multiple intents in one query.

Level 4 Indicates abilities to hold multiple conversations and comprehending complex human features.

Level 4a Capabilities to comprehend multiple input languages and responding in using different languages.

Level 4b Capabilities to comprehend feelings and sentiments and use it for responding.

The criteria for each level are divided into two main concerns as shown in table 4.3: understanding and response. Understanding refers to the level of comprehension the conversational AI system has and the type of natural language processing it can perform. Response refers to the system's response patterns and abilities to interact with the user. These two concepts make up the conversational AI's ability to interact and converse with a user.

Table 4.3 further specifies the features corresponding to each level, pinpointing how certain functionality enables the conversational maturity of a conversational AI platform. All relevant features have been introduced in section 4. This was done by analyzing the different features and how they correspond to the conversational maturity within the conversational AI platforms. To map the features to different maturity levels within the framework, we used an approach similar to what Venkatesh et al. [10] used in their paper of evaluating and comparing different conversational agents. The main difference between their approach and ours was that we mainly focused on mapping specific features to different maturity levels and analyzing entire conversational AI platforms, whilst they focused on whole concepts within conversational AI and how these concepts correlate to the quality of the conversational agents. We considered 5 main concepts that are of importance when evaluating the features that where relevant:

1. **Conversational comprehension** - how will the feature affect the conversational AI platform's ability to comprehend and understand the user [10].
2. **Conversational utterance** - how will the feature affect the ability to respond and interact with the user for a conversational AI platform [4].
3. **Conversational coherence** - how are the conversational skills and the related context of the conversation affected by the feature. Also how deep a conversation can get, the more information stored within one conversation the deeper the conversation can get [34].
4. **Conversational breadth** - is the conversation bound to specific topics or does the feature allow the conversation to expand beyond specific features, these topics can be anything from small-talking to telling jokes [10].
5. **Conversational emotion** - does the feature bring any specific human-like emotions to the conversation, such as using different linguistics to convey different feelings and using different tones for speech responses [35].

With these concepts in mind, we then plotted the specific features in order of conversational magnitude. Here we again used the guidelines set out by the human language frameworks CEFR, ILR and ACTFL. We considered what aspects make a good human conversation, as described by See et al and Clark et al [4][34]. By doing this we could plot the features with conversational maturity in mind within the different levels of the framework.

Below will be a set of examples to why some features were added to specific maturity levels. To achieve level 1a, a platform needs to understand and respond to simple one-shot queries and phrases, which requires support for simple *intent* and *entity* definition as well as basic *dialog management* based on a definition of isolated words. *Intent* and *entity* is one of the basic features for conversational AI platforms, as mentioned in section 4.1.2, and plays a crucial part in conversational AI. It also needs to be able to provide some basic *input* and *output processing*.

The more complex features a conversational AI platform encompasses the more likely it is to achieve a high conversational maturity level. One feature that significantly affects the conversational maturity of a conversational AI is *context*. In conjunction with other features, it plays a very crucial role in creating a more fluid conversation between the AI and the user. Having *context* would allow for more in-depth conversation topics and a conversation that will hold the user's engagement over a longer period of time. It was included from level 2b since it is one of the main concepts of conversations and the necessary features to support *context* reside within the lower levels.

Another complex feature of conversational AI is the ability to process and understand multiple intents within one query. This is a continuation of the basic *intent* feature, which only allows for a single intent per query. Thus making multiple intents a more complex feature, corresponding to level 3b of the framework. This feature would not only require all the aforementioned features, but also need features such as: *multiple intents per entity* and *nested intents* to fully behave as described in section 4.1.2. Making it possible to handle many different user requests at once makes for a more human-like conversation since the only limitation for how many questions a human can process lays in the memory of that human per se.

One of the most important and challenging factors of human conversation is the interpretation and understanding of emotions [36]. Emotions add a whole new element to conversations, this addition can change a whole conversation by simply using a different tone or changing the wording within a sentence. A conversational AI system that can both interpret and respond with feelings create a more natural and as such a more mature conversation, allowing the development of a proper relationship with the user [37, 18]. Though it might not be as advanced as human emotions yet, it still provides a new layer of maturity to a conversational AI platform. It is extremely hard for a conversational AI platform to analyze and read emotions if the communication is done through only textual input and output [4]. Thus, in the conversational platforms mentioned earlier this functionality is only supported if the features *sentiments* and *tone analysis* are ex-

istent. This together with features such as *context*, *multiple intents* and *multi-domains* make for a very complex platform that can support conversations of human maturity. This was therefore added as the final level of the maturity framework.

Table 4.3: Assessment framework for conversational AI platforms: maturity levels.

Level	Understanding	Response	Features
Level 1a	• Can understand simple one-shot queries, like yes-or-no questions and questions for previously defined names. • Can understand simple phrases like greetings or information regarding the domain at hand. • Can understand one intent per entity defined in the conversational AI.	• Can respond with isolated words and numbers like "yes", "no" and "50". • Can initiate a dialog with the user, instead of waiting for a user input.	• Intent • Entity • OneShotQueries • YesNoQuestions • InputModality • OutputModality
Level 1b	• Can understand queries that have been explicitly defined for the domain at hand. • Can also understand simple queries for related domains.	• Can build a short phrase or sentence using a few connected words, to produce and response.	• OpenQuestions • MultipleConversationDomains • DialogInitiation
Level 2a	• Can understand longer queries with nested intents within the defined domain. • Can understand queries for related domains.	• Can respond with full sentences containing more information. • Can use affirmation to confirm the users intent. • Can ask the user to repeat the query if the query wasn't understood or misheard.	• Affirmation • Rephrasing • FallbackActions • SearchOrientedDialog • SlotFilling

Table 4.3: Assessment framework for conversational AI platforms: maturity levels.

Level 2b	• Can understand the conversational context and keep that context memorized throughout a short conversation. • Can understand multiple intents for a single entity defined in the conversational AI. • Can comprehend propositionality.	• Will ask the user for additional information if there is missing information for the intent. • Can respond with follow up questions to further continue the conversation.	• Contextual-Dialogs • MemoryFor-Context
Level 3a	• Can understand context in a sentence and keep that context memorized throughout an entire conversation. • Can use spelling correction and understand policies and censorship. • Can comprehend small talking, like asking: "How are you?"	• Can respond with sentences regarding small talk, like: "I'm doing very good!".	• SpellingCorrection • SmallTalk
Level 3b	• Can understand multiple intents in one query. • Can separate between language-specific and non-language-specific information.	• Can respond with multiple answers to cover all intents in the query.	• MultipleUserIntents • Language-Separation

Table 4.3: Assessment framework for conversational AI platforms: maturity levels.

Level 4a	• Can shift between different contexts within the same conversation. • Can understand at least 2 input languages.	• Can have sentiments when answering queries to add a more human aspect. Can respond in different languages. • Can translate information for the user.	• TopicShifting • MultiLanguage • Sentiments • Policies
Level 4b	• Can understand the sentiments and feelings. • Can analyze the users' speech input by using linguistic analysis to detect emotion and language tones.	• Can convey feelings when responding to queries; such as anger, comfort and etc.	• ToneAnalyzer • VoiceActivityDetection

5
Discussion

The results of this study will be discussed in this chapter. The first sections will discuss the results and end with describing some of the limitations of this study. The last section will discuss the potential validity threats for this study.

5.1 Significance of the feature mapping and maturity framework

One of the main purposes of this study was to map out and summarize features of platforms aimed at developing conversational AI systems. The need for this type of study has been prompted by the recent increase in different platforms for this purpose. RQ1 was aimed towards identifying as many platforms as possible to create the largest possible mapping of features. We showed this mapping in section 4.

The existing literature described in section 2 generally describes the features of conversational AI in broader concepts such as dialog management and response generation, or by evaluating specific features such as the Natural Language Understanding capabilities of a certain platform. In creating this feature mapping it can bridge the more general concepts of conversational AI with the more specific features that they are built upon. Another benefit of this mapping is that it groups features with the same functionality under common names based on how they are described in both literature and what name is most common within the platforms. With this, we make a step towards improving the current situation, in which there is no agreed-upon standard for terminology [22]. This can help individuals who wish to compare different platforms and determine which features set them apart. It can also serve as a benchmark for future evaluation of new or updated platforms. The methods described in literature

for evaluating conversational AI is mostly concerned with the systems themselves and not the platforms they are created on [10]. They also describe metrics that are useful, but often broad and not directly connected to specific features. Examples of such metrics are engagement (how interesting a conversation is) and conversational user experience. The framework that we have created is focused on evaluating the maturity of the platforms given the features of that specific platform. Making it possible to get an overview of the conversational capabilities of a system created on that platform by analyzing the features that the platform supports.

A situation where the framework can be applied is when a developer is deciding on a particular platform to develop their system on. This decision is obviously very dependent on the environment the system is to be applied in. In certain scenarios, it might be enough with conversational AI that can understand very simple queries and have very simple responses. In others, it might be necessary to have AI that can understand very complex sentences and convey emotions. Not using the most mature platform might be driven by other factors such as developer experience and available time. Some of the platforms are easier to understand and manage without extensive knowledge of conversational AI. With this in mind, the trade-off between less complexity and lower conversational maturity might be worthwhile if the maturity does not have any major impact on the overall experience.

For developers and researchers the framework can be useful to guide their next steps. It gives an overview of the platforms current conversational maturity and suggests what features might be worth working towards to increase the human-like capabilities of the platform. It can also be useful for researchers that are looking to propose improvements to certain conversational AI platform technology. It can serve as a basis for justifying why the addition or improvement of a specific feature might increase the conversational ability of conversational AI systems.

5.2 Constructing and applying the conversational maturity framework

The process of creating the conversational maturity framework was divided into two parts, the first part was to structure and laying out the mapping of the framework. The second part was to define the framework using linguistics and features found within the analyzed platforms. These two parts together were done to aid the systematic evaluation of the conversational maturity level within conversational AI platforms (RQ3). One of the main problems with structuring a framework for conversational maturity within conversational AI platforms is that there is no current standard within the field. This makes it hard to define and state different maturity levels and how these maturity levels correlate with the features. Thus, we turned to human language development to find approaches and methods they use for constructing a framework. The feature model helped with finding a correlation between the feature and the conversational maturity of the platform. This greatly helped with the overall structure and linguistics for the conversational maturity framework.

The second problem arose in the second part of constructing the conversational maturity framework when trying to define the different levels of maturity. The main concern here was to use linguistics that is commonly used within the field of conversational AI and finding the most suitable definition for the different levels. By researching studies that have been done within conversational maturity and conversational AI, we could construct a framework that uses common terms and has a logical structure.

In table 5.1 below is a list of suggested levels for the analyzed conversational AI platforms. This table shows the suggested conversational maturity level of the platform and what features are missing for the platform to reach the next conversational maturity level. This assessment was done by looking at the feature matrix to see what features the different platforms support and then plotting these to the framework. The column with the features required for the next level can act as a suggestion for current developers of the platforms to specifically

focus on during development. This is also a way to indicate for the platform company on what features are necessary for the platform to reach a more human-like performance if this is the end goal of the platform.

Table 5.1: Conversational AI platform conversational maturity assessment

Platform	**Conversational maturity level**	**Features for next level**
DialogFlow	Level 3b	TopicShifting, Sentiments and Policies
Meya.ai	Level 1b	OpenQuestions and MultiConversationDomains
Microsoft Bot Framework	Level 3a	MultipleUserIntents
Houndify	Level 1b	SearchOrientedDialog
Amazon Lex	Level 3b	MultiLanguage and Policies
RASA	Level 2b	SpellingCorrection
IBM Watson Conversation	Level 3a	LanguageSeparation
VoiceXML	Level 3a	LanguageSeparation
Recast.ai	Level 1b	SearchOrientedDialog
Kore.ai	Level 1b	SearchOrientedDialog
AIML	Level 1b	SearchOrientedDialog
TDM	Level 2a	MemoryForContext

5.3 Threats to Validity

Validity threats are those factors that may have an impact on the correctness of the results. To mitigate the effects of these factors it is necessary to identify potential threats and build strategies that avoid them. Bratthall and Wohlin [38] describe four different types of validity threats, summarized below.

5.3.1 Construct Validity

Construct validity refers to what extent the measures studied reflect what the researchers think they are studying and what the research questions describe [38]. The main source for identifying conversational AI systems was through the systematic literature study. If the converge of papers is not broad enough important platforms could be missed. To increase the precision of empirical research Runesson and

Höst [39] suggest using triangulation. The process of triangulation involves taking many different approaches to a certain study question. Triangulation becomes increasingly useful when collecting qualitative data.

For this study, two different methods of triangulation were used: data triangulation and observer triangulation.

data triangulation was used as several different sources were used to identify conversational AI platforms. Database searches were done to find the initial source of papers. Snowballing then extended the number of papers from which conversational platforms were identified. A company in the conversational AI field also provided some platforms. Lastly searches through google search engine provided additional platforms.

observer triangulation was used since both researchers were involved in the selection of papers and the identification of features of the different platforms as well as the creation of the maturity framework. Additionally, the supervisors from both the company and the university reviewed the information that had been gathered which allowed for further fine-tuning, especially of the features used to create the framework.

Another major threat is that the study only considers the publicly available documentation for the different platforms. The documentation may not always contain all the information regarding the platform and its features. Furthermore, it might describe features which are not actually available in the platform. Consequently, the conversational maturity of some of the platforms could be assessed too high or too low. By only reading available documentation, we trust that the platform companies have stated the information correctly and that the descriptions are neither over- nor understated. However, some platform companies do not want to release all the information regarding the platform and its' features which is one of the reasons some platforms considered for this study were excluded. Other companies might overstate the contained features as part of a selling strategy.

Future work that can be conducted here is to gain access to more platforms and possibly to more complete documentation of the platforms used in this study and extend the feature mapping. From this, the researcher could also increase the maturity framework with any features that may be detrimental to achieve some level of maturity. To further increase the validity of this study it would be of interest to conduct testing for each individual platform to determine how their specific features impact the conversational maturity. These tests should be setup to determine what features are necessary for the lower levels of conversational maturity, but also be able to identify features regarding higher levels. It should for example contain cases that can test the conversational coherence, i.e. basic cases that check if the platform can handle conversations that do not rely on previous information and cases that determines if it supports features such as context that do allow for more coherent dialogs. Doing so would increase the evidence of the features impact on the platforms conversational maturity without solely relying on what is stated by the platform companies.

5.3.2 Conclusion Validity

Conclusion validity is related to how certain we can be that the treatment used in the study is related to the observed outcome [38]. Meaning how different the results would be if another researcher would conduct the same experiments. The main threats to conclusion validity in this study is that of identifying search terms. For the snowballing procedure, it is important that the initial start set covers as large an area of the research field as possible. Another researcher conducting this study could formulate different search terms and as such obtain different results. To reduce the threat all the search terms used through the study have been detailed in appendix A.

5.3.3 Internal Validity

Internal validity is considered when a variables causality is being examined. When a researcher is examining if a variable A is affecting an investigated variable B there may also be an unknown or uncontrollable variable C that affects the outcome [38]. For this study the authors identified one specific such threat: exclusion bias. During the

snowballing procedure, platform identification and literature review for maturity frameworks there could be a possibility of researcher bias in including or excluding a certain paper or platform. To mitigate this bias specific inclusion and exclusion criteria were setup based on the research questions [40].

5.3.4 External Validity

External validity refers to what extent one can generalize the findings of the study and what interest it holds to people outside the specific case investigated [38]. The authors could not identify any external validity threats of this study. The literature studied in this thesis covers a large number of different conversational AI concepts and platforms. The features that have been elicited are general within the entire field and is not specific to one case. The framework created is based on these features and concepts and could therefore be applied to any current or future conversational AI platform.

6
Conclusion

We present a conversational AI maturity framework for assessing conversational AI platforms, based on the ability of the produced systems to conduct conversations. By supporting the understanding of how the features of a conversational AI platform correspond to conversational ability, this framework can help both users with choosing and developers with developing a powerful conversational AI system. Our framework is inspired by related frameworks for human language development. Comparable to the way in which a human speaker learns a language, the levels of conversational maturity in our framework indicate the ability to conduct and engage in a natural conversation with a user.

Our framework is based on, and incorporates results from an analysis of the state-of-the-art conversational AI platforms, which we identified in a literature review. We considered the documentation of these platforms to extract their common and unique features, which we then grouped into a feature model to provide a high-level overview of all the different existing features. Each feature comes with a description to support the understanding of its use, context, and scope. We related the features to conversational maturity and used them to develop the maturity levels in our framework.

Our results show that the various existing conversational AI platforms share significant commonalities. In the future, to bridge different terminologies and support users in flexibility choosing a platform according to their current needs, one aim is to develop a domain-specific language together with code generators for the various platform. Such an infrastructure allows for developing a system on a high level, and transforming the specification into an implementation for a concrete platform. It can also support the migration between different plat-

forms when a platform with higher conversational maturity becomes available.

Bibliography

[1] R. Menon, "Council Post: The Rise Of Conversational AI." https://www.forbes.com/sites/forbestechcouncil/2017/12/04/the-rise-of-conversational-ai/#63d2a333b91b, 2017.

[2] "Messenger Platform," 2019.

[3] P. Tsai, "Data snapshot: AI Chatbots and Intelligent Assistants in the Workplace - Spiceworks," 2018.

[4] L. Clark, N. Pantidi, O. Cooney, P. Doyle, D. Garaialde, J. Edwards, B. Spillane, E. Gilmartin, C. Murad, C. Munteanu, V. Wade, and B. R. Cowan, "What makes a good conversation?: Challenges in designing truly conversational agents," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, 2019.

[5] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (foda) feasibility study," tech. rep., Carnegie-Mellon University Pittsburgh, Institute for Software Engineering, 1990.

[6] K. Czarnecki and U. W. Eisenecker, *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, 2000.

[7] D. Nevsuc, J. Kruger, S. Stuanciulsecu, and T. Berger, "Principles of feature modeling," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 62–73, ACM, 2019.

[8] K. Czarnecki and S. Helsen, "Feature-based survey of model transformation approaches," *IBM Syst. J.*, vol. 45, pp. 621–645, July 2006.

[9] S. Erdweg, T. van der Storm, M. Völter, M. Boersma, R. Bosman, W. R. Cook, A. Gerritsen, A. Hulshout, S. Kelly, A. Loh, *et al.*,

"The State of the Art in Language Workbenches," in *SLE*, 2013.

[10] A. Venkatesh, C. Khatri, A. Ram, F. Guo, R. Gabriel, A. Nagar, R. Prasad, M. Cheng, B. Hedayatnia, A. Metallinou, R. Goel, S. Yang, A. Raju, and A. Alexa, "On Evaluating and Comparing Conversational Agents," tech. rep.

[11] E. Michiels, "Modelling Chatbots with a Cognitive System Allows for a Differentiating User Experience," tech. rep., IBM, Belgium, 2017.

[12] L. Cuno Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, "The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms. Technical Report," tech. rep., Department of Pure and Applied Sciences Urbino, Italy, 2017.

[13] J. Gao, M. Galley, and L. Li, "Neural Approaches to Conversational AI Question Answering, Task-Oriented Dialogues and Social Chatbots," tech. rep., 2018.

[14] M. Canonico and L. De Russis, "A Comparison and Critique of Natural Language Understanding Tools," in *CLOUD COMPUTING 2018 : The Ninth International Conference on Cloud Computing, GRIDs, and Virtualization*, (Barcelona), pp. 110–115, IARIA, 2018.

[15] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "THE MICROSOFT 2017 CONVERSATIONAL SPEECH RECOGNITION SYSTEM," tech. rep., 2017.

[16] D. Braun, A. Hernandez Mendez, F. Matthes, and M. Langen, "Evaluating Natural Language Understanding Services for Conversational Question Answering Systems," in *Proceedings of the SIGDIAL 2017 Conference*, (Saarbrucken), pp. 174–185, Association for Computational Linguistics, 2017.

[17] D. Schnelle-Walka, S. Radomski, B. Milde, and C. Biemann, "NLU vs. Dialog Management: To Whom am I Speaking? JOINT: Joining Ontologies and Semantics Induced from Text View project Trust and Reputation Models View project,"

[18] M. Mctear, Z. Callejas, and D. Griol, *The Conversational Interface Talking to Smart Devices*. Springer International Publishing, 2016.

[19] S. Santhanam and S. Shaikh, "A Survey of Natural Language Generation Techniques with a Focus on Dialogue Systems - Past,

Present and Future Directions," jun 2019.

[20] A. Patil, K. Marimuthu, N. Rao, and R. Niranchana, "Comparative study of cloud platforms to develop a Chatbot," *International Journal of Engineering & Technology*, pp. 57–61, 2017.

[21] C. Massimo and L. De Russis, "A Comparison and Critique of Natural Language Understanding Tools," in *CLOUD COMPUTING 2018* (B. Duncan, Y. Woo Lee, and A. Olmsted, eds.), (Barcelona), pp. 110–115, 2018.

[22] M. Mctear, "Conversational Modelling for Chatbots: Current Approaches and Future Directions," tech. rep., Ulster University, 2018.

[23] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu, J. Nunn, B. Hedayatnia, M. Cheng, A. Nagar, E. King, K. Bland, A. Wartick, Y. Pan, H. Song, S. Jayadevan, G. Hwang, A. Pettigrue, and A. Prize, "Conversational AI: The Science Behind the Alexa Prize," tech. rep., 2017.

[24] B. A. Shawar and E. Atwell, "Different measurements metrics to evaluate a chatbot system," in *Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings*, (Rochester, NY), pp. 89–96, Association for Computational Linguistics, 2007.

[25] C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering," in *International Conference on Evaluation and Assessment in Software Engineering*, pp. 38:1–10, 2014.

[26] G. Bowen, "Document Analysis as a Qualitative Research Method," 2009.

[27] K. Lee, K. Kang, and J. Lee, "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering," tech. rep., Pohang University of Science and Technology, 2002.

[28] Council of Europe, "Common European Framework for Languages: Learning, teaching assessment," tech. rep.

[29] E. Swender, J. D. Conrad, and R. Vicars, "ACTFL proficiency guidelines 2012," tech. rep., American Council on the Teaching of Foreign Languages, Alexandria, 2012.

[30] "Iteragency Language Roundtable," 2019.

[31] D. L. Lange and J. Lowe, Pardee, "Grading Reading Passages According to the ACTFL/ETS/ILR Reading Proficiency Standard: Can It Be Learned?," tech. rep., 1987.

[32] D. E. Tschierner, D. O. Bärenfänger, and I. Wanner, "Assessing Evidence of Validity of Assigning CEFR Ratings to the ACTFL Oral Proficiency Interview (OPI) and the Oral Proficiency Interview by computers (OPIc)," tech. rep., Institute for Test Research and Development, Leipzig, 2012.

[33] L.-F. Huang, S. Kubelec, N. Keng, and L.-H. Hsu, "Evaluating CEFR rater performance through the analysis of spoken learner corpora," 2018.

[34] A. See, S. Roller, D. Kiela, and J. Weston, "What makes a good conversation? How controllable attributes affect human judgments," tech. rep., 2019.

[35] T. S. Polzin and A. H. Waibel, "Detecting Emotions in Speech," tech. rep., 1998.

[36] U. Gupta, A. Chatterjee, R. Srikanth, and P. Agrawal, "A Sentiment-and-Semantics-Based Approach for Emotion Detection in Textual Conversations," tech. rep., 2018.

[37] "The influence of empathy in human-robot relations," *International Journal of Human Computer Studies*, vol. 71, no. 3, pp. 250–260, 2013.

[38] L. Bratthall and C. Wohlin, "Understanding some software quality aspects from architecture and design models," in *the 8th International Workshop of Program Comprehension*, vol. 2000-Janua, (Limerick), pp. 27–34, 2000.

[39] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," 2008.

[40] S. Keele, "Guidelines for performing Systematic Literature Reviews in Software Engineering," tech. rep., 2007.

A

Appendix 1

Table A.1: Search strings used to find start set for snowballing.

1	Conversational AI
2	Chatbots
3	Chatbot language
4	Chatbot feature
5	Conversational AI language
6	Conversational AI systems
7	Conversational AI features
8	Conversational AI development
9	Conversational AI comprehension
10	Chatbot comprehension
11	Developing AI chatbots
12	Conversational Agents
13	Conversational platforms
14	Conversational AI platforms
15	Conversational AI development platforms
16	Chatbot development
17	Chatbot platforms
18	Conversational AI comparison
19	Conversational Agent development platforms
20	Dialog management
21	Dialog management in chatbots
22	Dialog management conversational AI
23	Mandatory chatbot features
24	Common chatbot features
25	Conversational AI characteristics
26	Chatbot characteristics
27	Mandatory conversational AI features
28	Dialog flow development
29	IBM watson development
30	Amazon lex development
31	Microsoft bot framework development
32	VoiceXML
33	VoiceXML conversational AI

Table A.2: Search strings used to find systems for the analysis.

1	Alternatives to *platform*
2	*platform* competitors
3	*platform* alternatives
4	Systems similar to *platform*
5	Systems like *platform*
6	*platform* like systems
7	Platforms like *platform*
8	Platforms similar to *platform*
9	Applications similar to *platform*
10	Applications like *platform*
11	*platform* like applications

Table A.3: Search strings used for literature review for creation of conversational maturity framework.

1	Common language framework
2	Human language framework
3	Language framework
4	Developing language framework
5	developing conversational AI language framework
6	Understanding language frameworks
7	Conversational AI language framework
8	Conversational AI language evaluation
9	Conversational AI understanding language
10	Conversational AI language levels
11	Conversational AI language comprehension
12	Natural language framework
13	Natural Language Evaluation
14	Natural Language Complexity
15	Natural Language Comprehension
16	Conversational AI Language Maturity
17	Conversational AI Maturity

B

Appendix 2

Table B.1: List of papers used as start set for snowballing and which systems were found.

Paper	**Systems found**
T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management", tech. rep., NIPS 2017 Conversational AI workshop, Long Beach, 2017	RASA
R. I. A. Bohus, Dan, "The RavenClaw dialog management framework: Architecture and systems", Computer Speech & Language, vol. 23, pp. 332–361, 2009.	VoiceXML, RavenClaw
J. Rouillard, "Web services and speech-based applications around VoiceXML", tech. rep., Université des Sciences et Technologies de Lille, Villeneuve d'Ascq Cedex, 2006	VoiceXML
P. Milhorat, S. Schlögl, G. Chollet, J. Boudy, A. Esposito, and G. Pelosi, "Building the next generation of personal digital assistants", ATSIP 2014: 1st International Conference on Advanced Technologies for Signal and Image Processing, pp. 458–463, 2014.	Siri, Amazon Alexa, Google Assistant
M. Mctear, "Conversational modelling for chatbots: current approaches and future directions", tech. rep., Ulster University, 2018	DialogFlow, Amazon Lex, Microsoft Bot Framework, IBM Watson Conversation

Continuation of Table B.1	
Paper	Systems found
P. Priya Angara, Towards a Deeper Understanding of Current Conversational Frameworks through the Design and Development of a Cognitive Agent. PhD thesis, Gandhi Institute of Technology and Management, 2018.	DialogFlow, Amazon Lex, Microsoft Bot Framework, IBM Watson Conversation, Wit.ai, RASA
R. Sarikaya, P. A. Crook, A. Marin, M. Jeong, J. P. Robichaud, A. Celikyilmaz,Y. B. Kim, A. Rochette, O. Z. Khan, X. Liu, D. Boies, T. Anastasakos, Z. Feizol-lahi, N. Ramesh, H. Suzuki, R. Holenstein, E. Krawczyk, and V. Radostev, "An overview of end-to-end language understanding and dialog management for personal digital assistants", tech. rep., Microsoft Corporation, Redmond, WA, 2016	Cortana
M. Mctear, Z. Callejas, and D. Griol, The Conversational Interface Talking to Smart Devices. Springer International Publishing, 2016	Amazon Lex, Microsoft LUIS, VoiceXML, API.ai, AIML, wit.ai, IBM Watson Conversation, TrindiKit/ Dipper, NextIt, Interactions and Nuance nina
C. Massimo and L. De Russis, "A Comparison and Critique of Natural Language Understanding Tools", in Cloud computing 2018 (B. Duncan, Y. Woo Lee,and A. Olmsted, eds.), (Barcelona), pp. 110–115, Researchgate, 2018	wit.ai, LUIS, IBM Watson Conversation, Amazon Lex, Recast.ai

Continuation of Table B.1	
Paper	Systems found
J. Gao, M. Galley, L. Li, G. Brain, C. Brockett, A. Celikyilmaz, Y. Cheng,B. Dolan, P. Fung, Z. Gan, S. Lee, J. Li, X. Li, B. Liu, A. Madotto, R. Ma-jumder, A. Papangelis, O. Pietquin, C. Quirk, A. Ritter, P. Smolensky, A. Sor-doni, Y. Song, H. Suzuki, W. Wei, T. Weiss, K. Yuan, and Y. Zhang, "Neural Approaches to Conversational AI Question Answering, Task-Oriented Dialogues and Social Chatbots", tech. rep., 2018	LUIS, DialogFlow, Amazon Lex, IBM Watson Conversation, Cortana, Alexa, XiaoIce, Replika, Zo, Ruuh, Bing QA, Satori QA

Table B.2: List of papers found in first iteration of snowballing.

Paper	Systems found
D. Braun, A. Hernandez Mendez, F. Matthes, and M. Langen, "Evaluating Natural Language Understanding Services for Conversational Question Answering Systems", in Proceedings of the SIGDIAL 2017 Conference, (Saarbrucken), pp. 174–185, Association for Computational Linguistics, 2017	IBM Watson, wit.ai, Amazon Lex
R. Kar and R. Haldar, "Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements", tech. rep., School of Computing Sciences and Engineering, VIT University, Vellore, India, 2016	wit.ai, Microsoft Bot Framework
M. F. Mctear, "Spoken Dialogue Technology: Enabling the Conversational User Interface", tech. rep., University of Ulster, Ulster, 2002	VoiceXML, CPK Generic Dialogue System Platform, CSLU toolkit, CU Communicator system, GULAN, IBM voice server, NLSA, NUANCE, Speech Mania, Vocalis SpeechWare
J. Rouillard and P. Truillet, "Enhanced VoiceXML", tech. rep., 2005	VoiceXML
R. Catizone, A. Setzer, and Y. Wilks, "State of the Art in Dialogue Management", tech. rep., 2002	TrindiKit, SUNDial

Continuation of Table B.2	
Paper	Systems found
D. Bohus and A. I. Rudnicky, "RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda", tech. rep., Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, 2003	TeamTalk, BusLine, RoomLine, LARRI, RavenClaw
S. Larsson, "User-initiated Sub-dialogues in State-of-the-art Dialogue Systems", in Proceedings of the SIGDIAL 2017 Conference, (Saarbrücken, Germany), pp. 17–22, Department of Philosophy, Linguistics and Theory of Science University of Gothenburg, Association for Computational Linguistics, 2017	Siri, API.AI, Houndify, Cortana, Alexa
E. Fast, B. Chen, J. Mendelsohn, J. Bassen, and M. Bernstein, "Iris: A Conversational Agent for Complex Tasks", tech. rep., Stanford University, 2017	Iris
L. Cuno Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, "The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms", tech.rep., Department of Pure and Applied Sciences, Urbino, Italy, 2017	AIML, ALICE, Alexa, Google assistant, Cortana, Samsung S voice, Anna by IKEA, CHARLIE, MOOCBuddy, Nombot, SUNDIAL

Continuation of Table B.2	
Paper	Systems found
L. Wanner, E. André, J. Blat, S. Dasiopoulou, M. Farrùs, T. Fraga, E. Kamateri,F. Lingenfelser, G. Llorach, O. Martínez, G. Meditskos, S. Mille, W. Minker,L. Pragst, D. Schiller, A. Stam, L. Stellingwerff, F. Sukno, B. Vieru, and S. Vrochidis, "KRISTINA: A Knowledge-Based Virtual Conversation Agent", in Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection (Y. Demazeau, P. Davidsson, J. Bajo, and Z. Vale, eds.), pp. 284–295, Springer, Cham, 2017	KRISTINA
P. A. Crook, A. Marin, V. Agarwal, K. Aggarwal, T. Anastasakos, R. Bikkula,D. Boies, A. Celikyilmaz, S. Chandramohan, Z. Feizollahi, R. Holenstein,M. Jeong, O. Z. Khan, Y.-B. Kim, E. Krawczyk, X. Liu, D. Panic, V. Radostev,N. Ramesh, J.-P. Robichaud, A. Rochette, L. Stromberg, and R. Sarikaya, "Task Completion Platform: A self-serve multi-domain goal oriented dialogue platform", in Proceedings of NAACL-HLT 2016 (Demonstrations), (San Diego, CA), pp. 47–51, Association for Computational Linguistics, 2016	VoiceXML, RavenClaw, ClippyScript, TCP
G. De Gasperis, I. Chiari, and N. Florio, "AIML Knowledge Base Construction from Text Corpora", in Artificial Intelligence, Evolutionar Computing and Metaheuristics (Janusz Kacprzyk and Xin She Yang, eds.), pp. 287–318, Springer, Berlin, Heidelberg, 2013	AIML
A. Patil, M. K, N. R. A, and N. R, "Comparative study of cloud platforms to develop a Chatbot,"International Journal of Engineering & Technology, vol. 6, p. 57, 6 2017	IBM Watson Conversation, chatfuel, Heroku, Kore, Amazon Lex

Table B.3: List of papers found in second iteration of snowballing.

Paper	Systems found
F. Morbini, K. Audhkhasi, K. Sagae, R. Artstein, D. Gan Can, P. Georgiou,S. Narayanan, A. Leuski, and D. Traum, "Which ASR should I choose for my dialogue system?", in Proceedings of the SIGDIAL 2013 Conference, (Metz), p. 394–403, Association for Computational Linguistics, 2013	DialogFlow, Siri, AT&T Watson
B. A. Shawar and E. Atwell, "Different measurements metrics to evaluate a chat-bot system", in Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, (Rochester, NY), pp. 89–96, Association for Computational Linguistics, 2007	AIML
G. Campagna, R. Ramesh, S. Xu, M. Fischer, and M. S. Lam, "Almond: The Architecture of an Open, Crowdsourced, Privacy-Preserving, Programmable Virtual Assistant", in Proceedings of the 26th International Conference on World Wide Web - WWW '17, (New York, New York, USA), pp. 341–350, ACM Press,2017	Almond

C
Appendix 3

Table C.1: List of feature descriptions. Written in bold is features that have underlying features and written in italics is abstract features. The features are in order of the feature model.

Feature name	**Description**
System	Features regarding the system and the supported tools and platforms.
Content	Features regarding the content of conversations that a system offers. A content of a conversation is i.e. “Booking” or “ Setting an alarm”.
ContentCatalogs	The system has in-built content catalogues to simplify the development of the conversational AI bot. These catalogs contain entities and intents that are common within the selected field.
MultipleConversation-Domains	The system is built with domains that is independent from one another. Thus making the system support multiple domains under one conversation.
Development	Features regarding the development process of the conversational AI systems.
ErrorFeedback	The system provides error messages upon a error occuring, which is used to give the developer some feedback on what went wrong in the system.
MultiProgramming-Language	The system has support for two or more programming languages.

Continuation of Table C.1	
Feature name	**Description**
PredefinedSlotTypes	The system has in-built slot types that are common in conversations. Typical examples for built-in slot types include e-mail adresses, phone numbers and ZIP codes.
SystemVersioning	The system supports iterative development of the system with versioning.
TrainingData	The system uses training data to generate a NLU model for the conversational AI system. This model is used for NLP in the system to understand the input.
CustomTrainingData	The system allows for the developer to choose what data set the developers wants to use for training the NLU model.
ProgrammingTools	The system has in-built tools to help developers by simplifying the development process.
DebugTool	Debug tools that are available within the system development that are readily available for the developers to use.
ModelEvaluation	Lets the developer evaluate the NLU model generated by the system, for analysis purposes such as: to see if it fits the companies purpose.
VisualisationTools	Tools to simplify for the developer using visual aids, these can vary from visualising the dialog tree to drag and drop boxes for entities and intents.
InputProcessing	Features regarding the processing of the user inputs.
AutomaticUnderstanding	The conversation is automatically processed using Natural Language Understanding to get the system to understand what the user says and writes.
MultiLanguage	The system has support for two or more languages.

Continuation of Table C.1	
Feature name	**Description**
Propositionality	Distinguishes semantic roles for different answers of same sort, e.g. “from X to Y” vs “to X from Y”.
SpellingCorrection	The system automatically corrects spelling mistakes to make it easier for it to understand what the user means. This has a threshold on how much it can auto correct just like typing on a phone.
LanguageRecognition	The system can automatically detect the language which the user is inputing. I.e. “Hola como te llamas?” will be detected as input in spanish.
Translation	The system can translate conversations to any supported language. This requires the system to have LanguageRecognition.
Interfaces	Features regarding the different interfaces supported by the system.
FrontendIntegration	The system allows for calls to be made to the frontend interface, i.e. make phonecalls or send text messages via the phone.
SocialPlatformSupport	The system is integratable with social platforms such as Facebook, Slack or Instagram. This allows the developer to create chatbots within the platforms directly.
WebIntegration	The system allows for webhooks and other calls to the web, i.e. google searches and pulling weather information from the web.
Conversation	Features regarding the conversational part of the system.
LanguageSeparation	The system can separate between language-specific and non-language-specific information, to simplify translation and multi language maintenance.

Continuation of Table C.1	
Feature name	**Description**
ConversationOutput	Features regarding the system outputs during conversations.
DialogInitiation	The system allows for the developer to set if the bot shall initiate conversation or if it shall wait for the user to initialise.
Policies	Policies are used for dialogs and conversation, to set restrictions and guidelines. These can be anything from restricted words to censorship.
Sentiments	Sentiments are similar to human emotions such as angry, sad and happy. The system allows for the developer to set a overall system sentiment.
Clarification	Features regarding clarification of the users input, to let the system know that the information received is not incorrect.
Affirmation	Affirmation is used by the system to confirm the intent of the user. Examples: (User) What is the weather like in New York? (Bot) Did you ask for the weather in New York?
FallbackActions	Fallback actions is a mechanism used by the system if it didn't understand what the user said/wrote. These actions can be something like: (Bot) I didn't understand, please try again.
Rephrasing	Rephrasing is used by the system to confirm the intent of the user. Examples: (Bot) Did you mean New York?
ConversationTypes	Different conversation types that the system has support for.

Continuation of Table C.1	
Feature name	**Description**
SearchOrientedDialog	The system organizes different slot types in groups to simplify searches and to be able to quantify how many hits the system got from a specific search. I.e. (User) Search for Johan. (Bot) There where 12 Johan's found.
SlotFilling	The system will ask follow up questions to statements that are missing information, for example: (User) What is the weather like today? (Bot) Where would you like to check the weather for?
SmallTalk	The system can hold a conversation in a subject where the user does not have a goal, for example: (User) I don't feel too good today. (Bot) I hope you will feel better later!
ContextualDialogs	The possibility to hold more complex dialogs that keep the context throughout the conversation, The system will remember what the user has said/written previously.
MemoryForContext	The system keeps conversation history to keep the context of the conversation, this memory size can vary depending on system.
TopicShifting	The system allows for multi-contextual dialogs, a dialog where the user switches between contexts. Allowing the user to converse about the weather whilst also conversing about the upcoming events in the area.
Questions	Features regarding the question part of the conversations.

Continuation of Table C.1	
Feature name	**Description**
MultipleUserIntents	The system allows for phrases/ sentences with multiple questions, usually supports no more than 3 questions in one phrase.
OneShotQueries	A user command that only requires an answer with no further conversation needed. I.e. (User) Tell me the time. (Bot) The time is 12:43.
OpenQuestion	The system supports for the use of open questions. Examples "What is the weather like?" & "How far is it to Stockholm?"
YesNoQuestion	The system supports questions that only require a yes or no answer from the bot.
DialogFlow	The dialog flow of the whole conversation and the nodes which it can go.
DialogDefinition	The developer can create their own nodes in the dialog tree, to let the developers create conversations to their needs.
Entity	An entity is the different options the system can prompt the user during a conversation. Entity is used to let the system know what steps to take the conversation forward.
EntityDefinition	Allows the developer to define their own entites. I.e. creating a new entity that is "I'm feeling good" for user questions like "How are you feeling?".
TrainingPhrases	Different user sentences matches one and the same intent. I.e. "What's the weather like?" and "How is the weather outside?" both being mapped to the intent: weather report.
NrOfIntents	The number of intents allowed within the system.

Continuation of Table C.1	
Feature name	**Description**
Intent	Intents are used to identify, what the user wants to know/ do using the system.
IntentDefinition	Allows the developer to define their own intents. I.e. creating a new intent that is "What time is it?" so that the system can understand the user if he/she asks for the time.
Synonyms	The system automatically takes into consideration to synonyms and phrases that have similar meaning. I.e. "large" and "extensive".
NrOfEntities	A restricted number of entities allowed within a system.
Speech	Features specific to speech and speech processing.
ToneAnalyzer	The system can analyse the tone of the user by using linguistic analysis to detect emotion and language tones.
VoiceActivityDetection	Voice activity detection is used for the system to understand when the user has stopped talking so that the system can shut off the microphone from listening any further.
InputModalities	The different input types supported by the system.
TextInput	The system allows for text as an input, a string of characters that the user writes.
SpeechInput	The system allows for speech as an input, this means recordings of audio through a microphone.
ImageInput	The system allows for image as an input, a image filetype such as .PNG, .JPEG, .TIFF, etc.

Continuation of Table C.1	
Feature name	**Description**
URLInput	The system allows for a URL as an input, an string that identifies to a websource on a network. This is used for redirection and lets the user choose webhook sources.
OutputModalities	The different output types supported by the system.
TextOutput	The system has support to output a text as a response to an user request.
SpeechOutput	The system has support to output a text-to-speech response to an user request if the user has a speaker available.
ListOutput	The system has support to output a list response to an user request, this list can contain options for the user to choose or simply a list of items that the user requested for.
ImageOutput	The system has support to output an image as a response to an user request.