

Computational Intelligence in Conversational UI, A BotLibre Case Study. A survey paper

Dr Bheemaiah, Anil Kumar , A.B, Seattle WA
98125
bheemaiah@alumni.iitm.ac.in

Abstract

This publication is on the computational intelligence of Chat conversational UIs that spans from, NLP, RPA, DNN's, emergent computing to other paradigms of A.I which also include rule based systems and heuristic definition languages(HDL). The design of the BotLibre conversational UIs is presented as a case study, including future work in improving the knowledge management and database use, from postgre SQL to NoSQL and more sophisticated database clusters.

Keywords: conversational UI, botlibre, NLP, Deep Learning, RPA.

Introduction

A conversational UI is a computer program or an artificial

intelligence which conducts a conversation via auditory or textual methods. Animated conversational UI are now common and include full or partial animations based on text.

conversational UI are typically used in dialog systems for various practical purposes including customer service or information acquisition.

Today, most conversational UI are either accessed via virtual assistants such as Google Assistant and Amazon Alexa, via messaging apps such as Facebook Messenger or WeChat, or via individual organizations' apps and websites.

The classic historic early conversational UI are ELIZA and PARRY. More recent notable programs include A.L.I.C.E., Jabberwacky and D.U.D.E. While ELIZA and PARRY were used exclusively to simulate typed conversation, many conversational UI now include functional features such as games and web searching abilities.

In 1984, a book called The Policeman's Beard is Half Constructed was published, allegedly written by the conversational UI Racter.

Some more recent conversational UI also combine real-time learning with evolutionary algorithms that optimise their ability to communicate based on each conversation held.

conversational UI competitions focus on the Turing test or m

ore specific goals.

conversational

UI have been used in applications such as customer service, sales and product education.(Bheemaiah, n.d.; Janarthanam 2017; McKay 2013; “Developing and Evaluating Conversational Agents” 2000)

Many companies' conversational UI run on messaging apps like Facebook Messenger, WeChat, WhatsApp, LiveChat, Kik, Slack, Line, Telegram, or simply via SMS. They are used for B2C customer service, sales and marketing.(Janarthanam 2017; Bheemaiah, n.d.; Oehlman and Blanc 2011)

In 2016, Facebook Messenger allowed developers to place conversational UI on their platform.

Previous generations of conversational UI were present on company websites, e.g. As kJenn from Alaska Airlines which debuted in 2008 or Expedia's virtual customer service agent which launched in 2011.(Verha gen et al. 2011)

The newer generation of conversational UI includes IBM Watson-powered "Rocky", introduced in February 2017 by the New York City-based e-commerce company Rare Carat to provide information to prospective diamond buyers.

A SaaS conversational UI business ecosystem has been steadily growing since the F8 Conference when Zuckerberg unveiled that Messenger would allow conversational UI into the app.(Bello 2019; Shevat 2017;

McTear, Callejas, and Griol 2016)

Hello Barbie is an Internet-connected version of the doll that uses a conversational UI provided by the company ToyTalk, which previously used the conversational UI for a range of smartphone-based characters for children.

The process of creating a conversational UI follows a pattern similar to the development of a web page or a mobile app.

The conversational UI design is the process that defines the interaction between the user and the conversational UI.

The conversational UI designer will define the conversational UI personality, the questions that will be asked to the users, and the overall interaction.

An important part of the conversational UI design is also centered around user testing.

The process of building a conversational UI can be divided into two main tasks: understanding the user's intent and producing the correct answer.

The second task may involve different approaches depending on the type of the response that the conversational UI will generate.

To keep conversational UI up to speed with changing company products and services, traditional conversational UI development platforms require ongoing maintenance.

To eliminate these costs, some startups are experimenting with Artificial Intelligence to develop self-learning conversational UI, particularly in Customer Service applications.

The process of building, testing and deploying conversational UI can be done on cloud based conversational UI development platforms offered by cloud Platform as a Service providers such as Yekaliva, Oracle Cloud Platform, Snatch conversational UI and IBM Watson.

These cloud platforms provide Natural Language Processing, Artificial Intelligence and Mobile Backend as a Service for conversational UI development.

There are a lot of API's available in case you are building your own conversational UI like these AARC. Malicious conversational UI are frequently used to fill chat rooms with spam and advertisements, by mimicking human behaviour and conversations or to entice people into revealing personal information, such as bank account numbers.

Natural Language Processing (Goldberg 2017; Liu 2012; Perez-Marín and Diana 2011) Perez-Marín et al. is a good survey on NLP algorithms in conversational UI design.

Natural language processing is a field of computer science, artificial intelligence concerned with the interactions between computers and human languages, and, in particular, concerned with programming computers to fruitfully process large

natural language data.

Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.

Some notably successful NLP systems developed in the 1960s were SHRDLU, natural language system working in restricted "Blocks worlds" with restricted vocabularies, and ELIZA, a simulation of a Rogerian psychotherapist, written by Joseph Weizenbaum between 1964 and 1966.

Starting in the late 1980s there was a revolution in NLP with the introduction of machine learning algorithms for language processing.

In recent years, there has been a flurry of results showing deep learning techniques achieving state-of-the-art results in many natural language tasks, for example in language modeling, parsing, and many others.

Since the so-called "Statistical revolution" in the late 1980s and mid 1990s, much Natural Language Processing research has relied heavily on machine learning.

Formerly, many language-processing tasks typically involved the direct hand coding of rules, which is not in general robust to natural language variation.

In languages such as Turkish or Meitei, a highly agglutinated Indian language such an approach is not possible, as each dictionary entry has thousands of possible word forms.

Some written languages like Chinese, Japanese and Thai do not mark word boundaries in such a fashion, and in those languages text segmentation is a significant task requiring knowledge of the

vocabulary and morphology of words in the language.

Natural language generation Convert information from computer databases or semantic intents into readable human language.

Natural language understanding Convert chunks of text into more formal representations such as first-order logic structures that are easier for computer programs to manipulate.

Natural language understanding involves the identification of the intended semantic from the multiple possible semantics which can be derived from natural language expression which usually takes the form of organized notations of natural languages concepts.

As a result, the Chomskyan paradigm discouraged the application of such models to language processing.

Procedures as a Representation for Data in a Computer Program for Understanding Natural Language.

Deep Learning

(Tur et al. 2018; Bertero, n.d.; Yan 2018) Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design and board game programs, where they have produced results comparable to and in

some cases superior to human experts.

Most modern deep learning models are based on an artificial neural network, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in DeepBelief Networks and Deep Boltzmann Machines.

Importantly, a deep learning process can learn which features to optimally place in which level on its own.

No universally agreed upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth > 2 .

For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation.

Deep learning algorithms can be applied to unsupervised learning tasks.

Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks.

The term Deep Learning was introduced to the machine learning community by Rina Dechter in 1986, and to Artificial Neural Networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons.

Many aspects of speech recognition were taken over by a deep learning method called long short-term memory, a recurrent neural network published by Hochreiter and Schmidhuber in 1997.

The 2009 NIPS Workshop on Deep Learning for Speech Recognition was motivated by the limitations of deep generative models of speech, and the possibility that given more capable hardware and large-scale data sets that deep neural nets might become practical.

Multiview deep learning has been applied for learning user preferences from multiple domains.

These developmental models share the property that various proposed learning dynamics in the brain support the self-organization somewhat analogous to the neural networks utilized in deep learning models.

"Realistically, deep learning is only part of the larger challenge of building intelligent machines. Such techniques lack ways of representing causal relationships have no obvious ways of performing logical inferences, and they are also still a long way from integrating abstract knowledge, such as information about what objects are, what they are for, and how they are typically used. The most powerful A.I. systems, like Watson use techniques like deep learning as just one element in a very complicated ensemble of techniques, ranging from the statistical technique of Bayesian inference to deductive reasoning."

Learning a grammar from training data would be equivalent to restricting the system to commonsense reasoning that operates on concepts in terms of grammatical production rules and is a basic goal of conversational UI human language acquisition and AI. As deep learning moves from the lab into the world, research and experience shows that artificial neural networks are vulnerable to hacks and deception.

Emergent Algorithm

An emergent algorithm is an algorithm that exhibits emergent behavior.

In essence an emergent algorithm implements a set of simple building block behaviors that when combined exhibit more complex behaviors.

One example of this is the implementation of fuzzy motion controllers used to adapt to conversational UI movement in response to environmental obstacles.

Other examples of emergent algorithms and models include cellular automata, artificial neural networks and swarm intelligence systems.

Conversational UI Process Automation

Conversational UI process automation is an emerging form of business process automation technology based on the notion of software conversational UIs or artificial intelligence workers.

In contrast, RPA systems develop the action list by watching the user perform that task in the application's graphical user interface, and then perform the automation by repeating those tasks directly in the GUI. This can lower the barrier to use of automation in products that might not otherwise feature APIs for this purpose.

RPA tools have strong technical similarities to graphical user interface testing tools.

RPA tools differ from such systems including features that allow data to be

handled in and between multiple applications receiving email containing an invoice, extracting the data, and then typing that into a bookkeeping system.

As a form of automation, the same concept has been around for a long time in the form of screen scraping but RPA is considered to be a significant technological evolution of this technique in the sense that new software platforms are emerging which are sufficiently mature, resilient, scalable and reliable to make this approach viable for use in large enterprises.

Such an illustration perhaps serves to demonstrate the level of intuition, engagement and ease of use of modern RPA technology platforms, that leads their users to relate to them as beings rather than abstract software services.

The "Code free" nature of RPA is just one of a number of significant differentiating features of RPA vs. screen scraping.

The hosting of RPA services also aligns with the metaphor of a software conversational UI, with each conversational UI instance having its own virtual workstation, much like a human worker.

The implementation of RPA in business enterprises has shown dramatic cost savings when compared to traditional non-RPA solutions.

According to Harvard Business Review, most operations groups adopting RPA have promised their employees that automation would not result in layoffs.

Some analysts proffer that RPA represents a threat to the Business Process Outsourcing industry.

The thesis behind this notion is that RPA will enable enterprises to "Repatriate" processes from offshore locations into local data centers, with the benefit of this new technology.

Academic studies project that RPA, among other technological trends, is expected to drive a new wave of productivity and efficiency gains in the global labour market.

Although not directly attributable to RPA alone, Oxford University conjectures that up to 35% of all jobs may have been automated by 2035.

In a TEDx talk hosted by UCL in London, entrepreneur David Moss explains that digital labour in the form of RPA is not only likely to revolutionise the cost model of the services industry by driving the price of products and services down, but that it is likely to drive up service levels, quality of outcomes and create increased opportunity for the personalisation of services.

Heuristics and HDL

(Shaw 1979; Pearl 2016)

In computer science, artificial intelligence, and mathematical optimization, a heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

A heuristic function, also called simply a heuristic, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow.

The objective of a heuristic is to produce a solution in a reasonable time frame

that is good enough for solving the problem at hand.

Results about NP-hardness in theoretical computer science make heuristics the only viable option for a variety of complex optimization problems that need to be routinely solved in real-world applications.

Execution time: Is this the best known heuristic for solving this type of problem? Some heuristics converge faster than others.

In some cases, it may be difficult to decide whether the solution found by the heuristic is good enough, because the theory underlying heuristics is not very elaborate.

One way of achieving the computational performance gain expected of a heuristic consists of solving a simpler problem whose solution is also a solution to the initial problem.

Such a heuristic is unable to find all the solutions to the initial problem, but it may find one much faster because the simple problem is easy to solve.

Initially, the heuristic tries every possibility at each step, like the full-space search algorithm.

In such search problems, a heuristic can be used to try good choices first so that bad paths can be eliminated early.

In their Turing Award acceptance speech, Allen Newell and Herbert A. Simon discuss the heuristic search hypothesis: a physical symbol system will repeatedly generate and modify known

symbol structures until the created structure matches the solution structure.

Each successive iteration depends upon the step before it, thus the heuristic search learns what avenues to pursue and which ones to disregard by measuring how close the current iteration is to the solution.

A heuristic method can accomplish its task by using search trees.

Heuristic scanning has the potential to detect many future viruses without requiring the virus to be detected somewhere, submitted to the virus scanner developer, analyzed, and a detection update for the scanner provided to the scanner's users.

To use a heuristic for solving a search or a knapsack problem, it is necessary to check that the heuristic is admissible.

Botlibre Design

BotLibre AI engine is somewhat more complex than many other AI and NLP engines.

conversational

UI Libre is a hybrid engine, in that it does not rely on a single technology.

Self Concept The conversational

UI Libre architecture is based on a concept of Self, modeled after the human brain.

conversational

UI Libre is an object oriented system, Java classes are used to model the concepts of the human brain.

Senses include Chat, HTTP, Twitter, Facebook,

Email, and IRC. Mood and Emotion The conversational UI's Mood and Emotions are newer concepts that we have added to the architecture.

The conversational UI's Mood processes input, which influence the state of the conversational UI's emotions.

Bot Libre vs AIML
BotLibre supports parsing and processing AIML, but has a much different architecture than typical AIML engines.

BotLibre actually does not process AIML internally, AIML is converted to conversational UILibre's scripting language Self.

The Self code is then compiled into a state machine and stored in the conversational UI's knowledge base.

Bot Libre can also store AIML patterns and templates in a less structured form in its knowledge base.

A key difference between BotLibre's architecture and other AIML engines is that conversational UI Libre deals with knowledge, whereas other AIML engines deal only with text.

In Bot Libre words are defined by their meanings, and meanings and their relationships can be used to process data and build responses.

BotLibre vs Neural Nets
BotLibre is not a neural net, but does share many similar concepts.

BotLibre's knowledge graph is similar to the structure of a neural net.

BotLibre's knowledge relationships each define a correctness factor that is increased or decreased when the system is corrected, or learns.

BotLibre vs NLP Natural Language Processors
typical parse language using a grammar and state machine, processing each word in the sentence until it is understood.

BotLibre Learning and Memory

(Summarized from BotLibre Technical Documentation.)

Overview

The learning and settings tab allows you to configure how your conversational UI learns and other settings. It gives several high level, and low level settings that let you customize how your conversational UI learns and thinks. This will influence how your conversational UI interacts with users, how it responds, how long it takes to respond, and how much memory it uses.

Learning Mode - The learning mode controls who your conversational UI will learn responses from. When enabled your conversational UI we learn every response to its response as a new response in context. Be careful enabling learning for service conversational UIs, as users can train your conversational UI to have bad responses.

Correction Mode - The correction mode controls who can correct your conversational UI's responses. Be careful enabling correction for service conversational UIs, as users can train your conversational UI to have bad responses.

Learning Rate - The learning rate is the % to increase a response's correctness when learning. Each time your conversational UI learns a new response to a question it will increase its correctness by this %. A response has a correctness from -100% to 100%. In conversation mode a conversational UI will use a response with a %50 correctness (by default). The default learning rate is 50%.

Script Timeout - The script timeout (in milliseconds) gives a limit to the amount of time for script processing. If a timeout occurs, the conversational UI will abort the script, and respond using response matching, or use a default response. This can be used to ensure the conversational UI does not take too long to give a response. The default is 10000 (10 seconds).

Response Timeout - The response timeout (in milliseconds) gives a limit to the amount of time the conversational UI will search for a matching response. When the conversational UI does not know a response for a question it will search for similar questions that it does know a response to. This can be used to ensure the conversational UI does not take too long to give a response. Smaller values make the conversational UI respond quicker, larger values can help the conversational UI find a better response. The default is 1000 (1 second).

Conversation Match - The conversation match % influences when the conversational UI will use a response in a conversation. If the response's correctness is less than the %, or for a response match, if the question's % match is less than the %, then the response will not be used. If no responses match the %, then the conversational UI will use a default response. The conversational UI is in conversation mode for 1v1 conversations, such as chat, privates, email, twitter mentions and direct messages. The default is 50%.

Discussion Match - The discussion match % influences when the conversational UI will use a response in a discussion. If the response's correctness is less than the %, or for a response match, if the question's % match is less than the %, then the response will not be used. If no responses match the %, then the conversational UI

will not respond. The conversational UI is in discussion mode for chat room conversations, such as chat rooms, IRC, Twitter status updates and searches. A chat room message that mentions the conversational UI's name is treated as a conversation message, not a discussion message. The default is 90%.

Enable Emoting - Configures the ability for users to teach conversational UI emotes. If disabled, only administrators will be able to teach conversational UI emotes. An emote associates an emotion with a word or phrase and influences the conversational UI's avatar and mood.

Enable Emotions - Configures the ability for the conversational UI feel or associate emotions. Disabling emotions can improve the conversational UI's performance somewhat, and prevent it from becoming self aware.

Allow JavaScript - Allow the conversational UI's responses to contain JavaScript. Caution should be used it enabling JavaScript to prevent security issues. For security reasons JavaScript cannot be enabled if learning is enabled. JavaScript is only allowed for commercial accounts.

Enable Comprehension - Configure if the conversational UI should attempt to identify language rules from conversations. Comprehension allows the conversational UI to self learn template or scripted responses, such as learning to

count, or inferring learned phrases like 'What is your name?' -> 'I am Jim' as `Template('I am {speaker}')`. When enabled comprehension will enable the conversational UI to extend its last script with its own code. Disabling comprehension can improve performance, and ensure the bot only responds exactly as you train it.

Enable Consciousness - Configure if the conversational UI should have a consciousness and temporal awareness. The consciousness is used to determine the best response, or best word or meaning based on the context. Objects increase their consciousness level based on their relationship input, and fade over time. Temporal awareness associates a timeframe and temporal order for input. Disabling the consciousness can improve performance.

Enable Wiktionary - Configure if the conversational UI should look up word definitions on Wiktionary. This helps the conversational UI identify names, nouns, verbs, adjectives, synonyms, antonyms, and word definitions. This is used by many of the bootstrap scripts such as NounVerbAdjective and WhatIs. Currently only the English Wiktionary is used. This can be disabled for non-English conversational UIs, or to improve performance and reduce memory consumption.

Enable Response Matching - Configure if the conversational UI should search for

similar questions and responses when encountering a question it does not know a response to. This heuristic can also be influenced using the conversation/discussion match %. Responses can also be given keywords, required words, previous and topics to improve response matching.

Check Exact Match First - Configure if the conversational UI should reply to questions with a known response before executing its scripts. This lets learned responses override scripted responses, and can improve response times for known responses.

Split Paragraphs - Configure if multi sentence inputs should be split up and processed as multiple input. This means your conversational UI's response will contain responses to each sentence in the users input. Some scripts may require this to be disabled to process * patterns.

Fix Case for Template Responses - Configure if template formula responses should be fixed to use proper case. When enabled the first word will be capitalized, and other words other than names will be lower case.

Learn Grammar - Configure if word associations and grammar should be learned. When enabled words will be associated with what words come before and after them. This helps the conversational UI choose the correct word for verbs and pronouns. This can be

disabled to improve performance, or avoid the conversational UI learning bad grammar from users.

Synthesize Response - Configure if a synthesized response should be used by default. This will have the conversational UI generate a unique response to the question based on the question's words. A synthesized response will only be used if the conversational UI has no response match, and has no default response. Learn grammar should be enabled for this feature.

Properties

Properties	Description
Learning Mode	Configure which type of users that conversational UI should learn from.
Correction Mode	Configure which type of users are allowed to correct the conversational UI's responses.
Learning Rate %	The % rate to increase a response's correctness when learning.
Script Timeout	Number of milliseconds to allow for script processing.

Response Timeout	Number of milliseconds to allow for response matching.	(disable to improve performance).
Conversation Match %	The % confidence required for a conversational UI to use or match a response in a conversation.	Enable Wiktionary Configure if the conversational UI should look up word definitions on wiktionary (English only, disable to improve performance).
Discussion Match %	The % confidence required for a conversational UI to use or match a response in a discussion.	Enable Response Matching Configure if the conversational UI should search for similar questions and responses when encountering a question it does not know a response to.
Enable Emotions	Config if users are allowed to associate emotions with responses.	Check Exact Match First Configure if the conversational UI should reply to questions with a known response before executing its scripts.
Enable Emotions	Config if the conversational UI should experience emotions.	Split Paragraphs Configure if multi sentence inputs should be split up and processed as multiple input.
Allow JavaScript	Allow the conversational UI's responses to contain JavaScript.	Fix Case for Template Responses Configure if template formula responses should be forced to use proper case.
Enable Comprehension	Configure if the conversational UI should attempt to identify language rules from conversations (disable to improve performance).	Learn Grammar Configure if word associations and grammar should be learned.
Enable Consciousness	Configure if the conversational UI should have a consciousness and temporal awareness	

Synthe Configure if a synthesized
size response should be used if no
Respon default response.
se

Conclusions and Future Work

We have thus described the computational intelligence and the state machines that define the conversational UILibre, conversational UIs Artificial Intelligence, this includes, consciousness, learning and knowledge data structures and the training mechanisms and the NLP mechanisms, RPA and emergent A.I are missing in conversational UILibre and can be added in Self scripts, KM can be improved from SQL to more advanced database clusters, which is a topic of a future publication.

Bello, Manasés. 2019. "Cloud-Based Conversational Agents for User Acquisition and Engagement." *Proceedings of the 9th International Conference on Cloud Computing and Services Science*.
<https://doi.org/10.5220/0007766105280534>.

Bertero, Dario. n.d. "Conversational Humor Recognition and Generation through Deep Learning."
<https://doi.org/10.14711/thesis-991012698866103412>.

Bheemaiah, Anil Kumar. n.d. "Personality In Conversational UI Design."
<https://doi.org/10.31224/osf.io/d3t8x>.

———. n.d. "The Universality of the Conversational UI Interface with Media in UX Design."
<https://doi.org/10.31224/osf.io/yw6d9>.

"Developing and Evaluating Conversational Agents." 2000. *Embodied Conversational Agents*.
<https://doi.org/10.7551/mitpress/2697.003.0013>.

Goldberg, Yoav. 2017. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Janarthanam, Sridhar. 2017. *Hands-On Chatbots and Conversational UI Development: Build Chatbots and Voice User Interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills*. Packt Publishing Ltd.

Liu, Bing. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

McKay, Everett N. 2013. "UI Design Examples." *UI Is Communication*.
<https://doi.org/10.1016/b978-0-12-396980-4.00006-8>.

McTear, Michael, Zoraida Callejas, and David Griol. 2016. *The Conversational Interface: Talking to Smart Devices*. Springer.

Oehlman, Damon, and Sébastien Blanc. 2011. "Mobile UI Frameworks Compared." *Pro Android Web Apps*.
https://doi.org/10.1007/978-1-4302-3277-3_11.

Pearl, Cathy. 2016. *Designing Voice User Interfaces: Principles of Conversational Experiences*. "O'Reilly Media, Inc."

Perez-Marin, and Diana. 2011. *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices: Techniques*

- and Effective Practices*. IGI Global.
- Shaw, Mildred L. G. 1979. "Conversational Heuristics for Eliciting Shared Understanding." *International Journal of Man-Machine Studies*.
[https://doi.org/10.1016/s0020-7373\(79\)80012-7](https://doi.org/10.1016/s0020-7373(79)80012-7).
- Shevat, Amir. 2017. *Designing Bots: Creating Conversational Experiences*. "O'Reilly Media, Inc."
- Tur, Gokhan, Asli Celikyilmaz, Xiaodong He, Dilek Hakkani-Tür, and Li Deng. 2018. "Deep Learning in Conversational Language Understanding." *Deep Learning in Natural Language Processing*.
https://doi.org/10.1007/978-981-10-5209-5_2.
- Verhagen, Tibert, Jaap van Nes, Joannes Franciscus Maria Feldberg, and Willemijn Maria Dolen. 2011. *Virtual Customer Service Agents: Using Social Presence and Personalization to Shape Online Service Encounters*.
- Yan, Rui. 2018. "'Chitty-Chitty-Chat Bot': Deep Learning for Conversational AI." *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.
<https://doi.org/10.24963/ijcai.2018/778>.