

Towards a Framework for Classifying Chatbots

Daniel Braun^a and Florian Matthes

Technical University of Munich, Department of Informatics, Munich, Germany

Keywords: Chatbot, Conversational Interface, Natural Language Interface, Classification Framework.

Abstract: From sophisticated personal voice assistants like Siri or Alexa to simplistic keyword-based search bots, today, the label “chatbot” is used broadly for all kinds of systems that use natural language as input. However, the systems summarized under this term are so diverse, that they often have very little in common with regard to technology, usage, and their theoretical background. In order to make such systems more comparable, we propose a framework that classifies chatbots based on six categories, which allow a meaningful comparison based on features which are relevant for developers, scientists, and users. Ultimately, we hope to support the scientific discourse, as well as the development of chatbots, by providing an instrument to classify and analyze different groups of chatbot systems regarding their requirements, possible evaluation strategies, available toolsets, and other common features.

1 INTRODUCTION

In 1950, Alan Turing famously described what we would call today a “chatbot” as part of his “Imitation Game” (Turing, 1950), now better known as Turing Test. However, it still took 16 more years until ELIZA (Weizenbaum, 1966) was developed, which is today widely regarded as the first chatbot (Jia, 2009; Abdul-Kader and Woods, 2015; Kerlyl et al., 2007).

The Term “chatbot” itself was not coined until 1994, another 28 years later, when Michael Mauldin referred to such programs like ELIZA as “ChatterBots” (Mauldin, 1994), because one could have informal conversations with them (chats).


In the same year that ELIZA was released, Ryan et al. described a system with a “Conversational Interface” (Ryan et al., 1966). Like ELIZA, their system used a dialogue with the user as input, but unlike “ChatterBots”, the system was focused on a very narrow domain instead of general conversations. For decades, the distinction between Conversational Interfaces and chatbots remained. Today, however, this distinction has mostly vanished outside of very specialised scientific communities.

From sophisticated personal voice assistants like Siri or Alexa to simplistic keyword-based search bots, the label “chatbot” is used broadly for all kinds of systems that use natural language as input. Recent

advances in natural language processing and machine learning, as well as the creation of central platforms, like Facebook and Telegram, sparked a new hype around chatbots, which lead to rapid developments within the industry, sometimes outpacing the progress within the scientific community. In some areas, this led to a lack of theoretical foundation.

One of them is the classification of chatbots. Since even the distinction between conversational interfaces and chatbots seems to have vanished, we are now lacking means to comprehensively classify conversational systems. For scientists, such a classification is important to compare and evaluate systems. A conversational interface which is focused on one specific task has to be evaluated differently than a classical “ChatterBot”.

For practitioners, a classification can help to define requirements and select the right tools. A keyword-based search interface has different requirements than a voice-commanded personal assistant. In this paper, we propose a framework for classifying chatbots alongside features which are meaningful to different stakeholders, in order to improve the development of chatbots and connect industry and research again more strongly.

^a  <https://orcid.org/0000-0001-8120-3368>

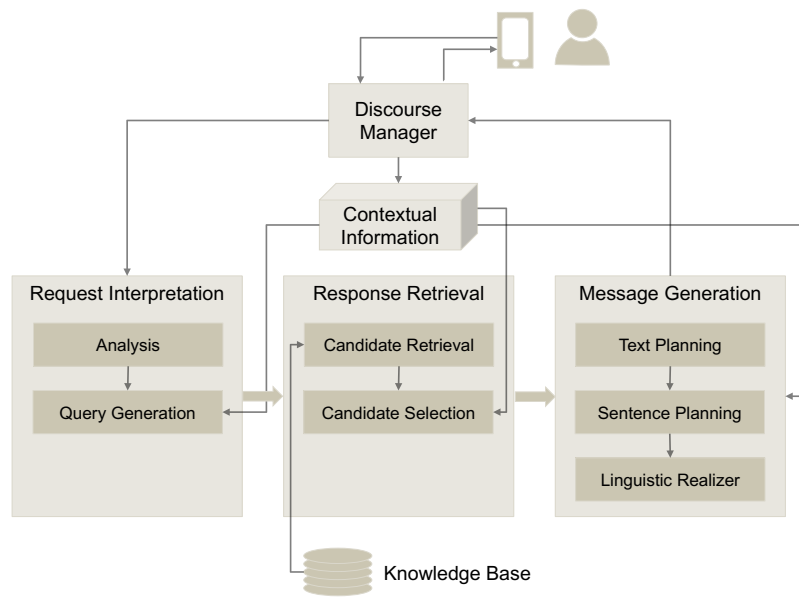


Figure 1: Reference architecture for chatbots (Braun et al., 2017).

2 METHODOLOGY

In order to develop a classification framework that is really relevant for the different stakeholder groups, we followed a design science research approach (Hevner, 2007; Hevner and Chatterjee, 2010). The main artefact we developed is the classification framework described in Section 4. Our research contributes to all cycles according to Hevner’s three-cycle view of design science (Hevner, 2007; Hevner et al., 2004).

In the previous section, we already explained how such a framework would be beneficial for different involved stakeholders (Relevance Cycle). The result of the Design Cycle is the proposed framework, which we also briefly evaluate in this paper by applying it to different systems and evaluating the resulting classification. Last but not least, our research is grounded on scientific literature as well as publications from industry, and informal expert interviews.

We hope that the developed framework will be an appreciated contribution to the knowledge space of chatbots and conversational interfaces and will lead to a more differentiated engagement with the subject (Rigor Cycle).

3 RELATED WORK

While, to the best of our knowledge, there have been no attempts to develop an overarching framework to classify chatbots based on all relevant features, differ-

ent researchers focused on individual aspects of chatbots in order to compare and classify them.

Shawar and Atwell e.g. compared “different measurements metrics to evaluate a chatbot system” (Shawar and Atwell, 2007). These included dialogue efficiency metrics, dialog quality metrics, and user satisfaction metrics. All measurements which are relevant for both, scientists and developers, but only indirectly for users.

Gianvecchio et al. proposed a classification system in order to find out whether a given chat partner is a chatbot or not (Gianvecchio et al., 2011) by taking into accounts factors like message size and inter-message delay. A classification that is mainly relevant for scientists.

On a more technical level, and therefore mainly relevant for developers, Abdul-Kader and Woods presented a survey on chatbot design techniques by analysing 25 Loebner prize winning chatbots (Abdul-Kader and Woods, 2015). Braun et al. evaluated Natural Language Understanding services for chatbots, including services from IBM, Microsoft, Google, and Facebook, and compared their classification quality (Braun et al., 2017). Moreover, they also suggested a reference architecture for chatbots shown in Figure 1.

4 PROPOSED FRAMEWORK

The framework we propose classifies chatbots based on six different categories. In this section, we will

Table 1: Application of the classification framework to three different chatbot systems.

System	Classification Framework														
	Domain	I/O		Timing		Flow		Platform			Understanding				
		Voice	Text	Synchronous	Asynchronous	Sequential	Dynamic	Messenger	Social Media	Standalone	Notifications	Keywords	Contextual	Personalized	Autonomous
Siri	open	X		X			X			X				X	
Mobility	mobility		X	X			X	X					X		
Mildred	flights		X	X			X	X					X		
Everyword	dict		X		X	X			X		X				

describe these categories and why they are relevant. Domain: The most decisive differentiator of chatbots is the domain they operate on. On the one hand, there are open domain systems like Alexa, Siri, and ELIZA, on the other hand, there are systems which are restricted to one or multiple domains, like weather reports or making a reservation. This distinction revives what used to be the differentiation between chatbots and conversational interfaces but sharpens it at the same time, by not only distinguishing between open and domain-specific but also between the actual domains. As mentioned before, for scientists this is e.g. important with regard to the evaluation of a system.

• **Input / Output:**

This category differentiates between voice and texts as input/output. This is especially relevant with regard to technology choices, but also with regard to the situations in which a chatbot can be used (e.g. hands-free).

• **Timing:**

We also differentiate between synchronous and asynchronous systems. If we talk to a voice assistant or chat with a bot on Telegram we expect an “immediate” answer, i.e. within seconds. From a Twitter bot we might not necessarily expect that. Synchronicity mostly imposes challenges regarding the performance of a system.

• **Flow:**

The flow of a conversation, i.e. the order in which messages are sent, can be either sequential or dynamic. A sequential chatbot for booking flights would e.g. expect to first receive the departure air-

port, then the destination, and then the time, while a dynamic system could process this information in an arbitrary order.

• **Platform:**

In their early days, chatbots were mostly standalone applications. Today, most chatbots are part of a messenger or social platform. Only voice assistants like Amazon’s Alexa are still commonly standalone applications or even separate devices.

• **Understanding:**

How “smart” a chatbot is perceived by its users largely depends on what input the system is able to understand. The “most stupid” systems do not process any input at all, they are merely a notification system which sends messages. More complex systems rely on keyword detection, even more sophisticated ones take into account the context of a conversation, i.e. they consider previous messages when interpreting new messages.

State-of-the-art systems take information about the user from previous conversations or other sources into account. Thus, over time, they become more personalized to the needs of their users. Beyond this, there would be chatbots which are fully autonomous and not just communicate with humans but also with other bots. However, we do not expect to see such kinds of systems any-time soon.

5 APPLICATION

In order to perform a first evaluation, or rather a “sanity check”, of our framework, we applied it to classify four very different chatbot systems: Apple’s Siri, a Telegram chatbot for intermodal mobility (Braun et al., 2018), Mildred, a Facebook chatbot from Lufthansa for online flight information (Sieber, 2019), and everyword, a Twitter bot that tweeted every word in the English language. The results of the classification are shown in Table 1.

The results show that the framework is able to cover the wide variety of the four bots. From a practitioners perspective, the classification gives a quick and comprehensive overview of the requirements of the different systems. In order to implement a bot like Siri, one would e.g. need voice-to-text and text-to-voice systems, as well as enough computing power to handle incoming requests synchronously. On the other end of the scale, everyword does not need any NLU capabilities at all and no powerful servers, but a connection to social media services.

From a scientific point-of-view, the classification would e.g. help to decide how to evaluate the different systems (e.g. Turing test for Siri, task-based evaluation for the Telegram and Facebook bot, and a questionnaire-based evaluation for everyword).

From the user perspective, the evaluation e.g. clarifies which types of input can be processed by the chatbot and on which platforms it can be used.

Obviously, this is not a full-fledged evaluation and the framework itself is still research in progress. Nevertheless, we hope that discussing our proposal with the scientific community will help us to improve it and adapt it to the needs of potential users.

6 AVAILABLE TOOLSETS

As mentioned before, we believe that a classification framework, like the one presented in this paper, has the power to help developers and scientists to make more educated decisions about which tools to use to develop a chatbot. In this section, we provide some examples, how a classification in the proposed framework could correlate with the choice of existing development tools.

6.1 RASA

RASA is a set of two open source libraries covering natural language understanding and dialogue management (Bocklisch et al., 2017).¹ Table 2 shows which

¹<https://rasa.ai>

requirements from the proposed classification framework can be implemented using RASA. It shows that RASA can e.g. handle textual input directly, but not spoken language.

Table 2: Capabilities provided by RASA.

Tool	Requirements	Impl.	
RASA	I/O	Voice	
		Text	X
	Timing	Synchronous	X
		Asynchronous	
	Flow	Sequential	X
		Dynamic	X
	Platform	Messenger	X
		Social Media	
		Standalone	X
	Understanding	Notifications	
		Keywords	X
		Contextual	X
		Personalised	
Autonomous			

Obviously, this does not mean that it is not possible to build voice-based chatbots with RASA, but it indicates that some additional tool will be necessary in order to build a chatbot that can be classified as voice-based.

6.2 Kaldi

Table 3: Capabilities provided by Kaldi.

Tool	Requirements	Impl.	
Kaldi	I/O	Voice	X
		Text	
	Timing	Synchronous	
		Asynchronous	
	Flow	Sequential	
		Dynamic	
	Platform	Messenger	
		Social Media	
		Standalone	
	Understanding	Notifications	
		Keywords	
		Contextual	
		Personalised	
Autonomous			

Kaldi² is an open source speech recognition library, i.e. Kaldi converts voice to text. (Povey et al., 2011) As shown in Table 3, the library does not provide any NLU capabilities whatsoever, in combina-

²kaldi-asr.org

tion with RASA however, it can help to build chatbots that accept voice input.

The example of Kaldi already shows that the proposed classification framework, which was designed to classify whole systems, rather than single components, is only partially applicable for this new task since it cannot capture that Kaldi is only able to process voice input, but not to produce voice output.

6.3 Chatfuel

Besides software libraries, online services are nowadays important tools for developers too. An example of an online service for the creation of chatbots is Chatfuel.³ Chatfuel provides a WYSIWYG interface which allows users to create end-to-end chatbots without any programming skills. Given this end-to-end approach, it is not surprising that Chatfuel implements a lot of the classes from the classification framework, as shown in Table 4

Table 4: Capabilities provided by Chatfuel.

Tool	Requirements	Impl.	
C.f.	I/O	Voice	
		Text	X
	Timing	Synchronous	X
		Asynchronous	X
	Flow	Sequential	X
		Dynamic	X
	Platform	Messenger	X
		Social Media	
		Standalone	
	Understanding	Notifications	X
		Keywords	X
		Contextual	X
		Personalised	
	Autonomous		

7 FUTURE WORK

In the future, we would like to link the framework with technical requirements that arise from the characteristics of different chatbot systems. That could help software engineers and architects to elicitate requirements for a chatbot system before building the system.

Moreover, based on the classification, (open source) software components or services could be suggested which have proven to be helpful in fulfilling the requirements imposed by the identified char-

³<https://chatfuel.com/>

acteristics of a chatbot, as already briefly shown in Section 6.

For existing systems, the classification framework could be linked to evaluation strategies which could help to conduct more meaningful and comparable evaluations of chatbots.

Moreover, it would be desirable to pay more attention to the needs of users as stakeholders and further investigate how a classification framework can help users to pick the right service for their needs.

ACKNOWLEDGEMENTS

This work has been sponsored by the German Federal Ministry of Education and Research (BMBF) grant A-SUM 01IS17049.

REFERENCES

- Abdul-Kader, S. A. and Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7).
- Bocklich, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.
- Braun, D., Hernandez-Mendez, A., Faber, A., Langen, M., and Matthes, F. (2018). Customer-centred intermodal combination of mobility services with conversational interfaces. In Drews, P., Funk, B., Niemeyer, P., and Xie, L., editors, *Multikonferenz Wirtschaftsinformatik (MKWI) 2018*, volume 4. Leuphana Universität Lüneburg.
- Braun, D., Hernandez-Mendez, A., Matthes, F., and Langen, M. (2017). Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIG-dial Meeting on Discourse and Dialogue*, pages 174–185.
- Gianvecchio, S., Xie, M., Wu, Z., and Wang, H. (2011). Humans and bots in internet chat: measurement, analysis, and automated classification. *IEEE/ACM Transactions on Networking (TON)*, 19(5):1557–1571.
- Hevner, A. and Chatterjee, S. (2010). Design science research in information systems. In *Design research in information systems*, pages 9–22. Springer.
- Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4.
- Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly*, 28(1):75–105.

- Jia, J. (2009). Csiec: A computer assisted english learning chatbot based on textual knowledge and reasoning. *Knowledge-Based Systems*, 22(4):249–255.
- Kerlyl, A., Hall, P., and Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. In *Applications and Innovations in Intelligent Systems XIV*, pages 179–192. Springer.
- Mauldin, M. L. (1994). Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *AAAI*, volume 94, pages 16–21.
- Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The kaldi speech recognition toolkit. Technical report, IEEE Signal Processing Society.
- Ryan, J. L., Crandall, R. L., and Medwedeff, M. C. (1966). A conversational system for incremental compilation and execution in a time-sharing environment. In *Proceedings of the November 7-10, 1966, fall joint computer conference*, pages 1–21. ACM.
- Shawar, B. A. and Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system. In *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies*, pages 89–96. Association for Computational Linguistics.
- Sieber, A. (2019). Dialogsysteme in der praxis. In *Dialogroboter*, pages 79–127. Springer.
- Turing, A. (1950). Computing machinery and intelligence—am turing. *Mind*, 59(236):433–460.
- Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.