

Virtual Customer Service: Building your Chatbot

David N. Sousa
Information Systems Department
University of Minho
Braga, Portugal
david.ns@hotmail.com

Miguel A. Brito
Algoritmi Centre
Information Systems Department
University of Minho
Braga, Portugal
mab@dsi.uminho.pt

Carlos Argainha
PrimaveraBSS
Braga, Portugal
carlos.arginha@primaverabss.com

ABSTRACT

Nowadays, there is a significant evolution in the simulation of conversations with humans through chats. These chatbots provide an improvement in organizational efficiency and enhance the performance in supporting customers and partners. That is why they are increasingly a trend on the web. There are several solutions to conversational virtual agents and chatbots at this time. Many of these solutions have built-in artificial intelligence, natural language processing, and understanding systems. This research aims to build a roadmap for those who want to develop a virtual assistant using a framework or a pre-built solution. This work compares the several products related to chatbots providing information that helps one to find the best solution for each case.

CCS Concepts

• **Human-centered computing**→**Human computer interaction (HCI)** →**Interaction paradigms**→**Natural language interfaces.**

Keywords

Chatbot; virtual assistant; conversational virtual agent; customer relationship management; CRM

1. INTRODUCTION

The increasing costs of the resources spent to support clients are an actual concern and justify some effort in order to release some pressure from the support team. From our personal experience, users do not read manuals nor search for information in FAQs. They simply jump to the phone or chat and ask questions like “what do I need to do, to create an invoice?”.

At the moment, there is a significant evolution in the simulation of conversations between humans through chats. These chatbots provide an improvement in organizational efficiency and increase the performance in supporting customers and partners. That is why they are increasingly a trend on the web.

On the other hand, when an enterprise wants to implement its own chatbot, it is far from obvious how to get a good implementation of it. From our research (see further below) there is no easy-to-use commercial off-the-shelf software. The majority of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICBIM 2019, September 12–14, 2019, Paris, France

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7232-9/19/09...\$15.00

DOI: <https://doi.org/10.1145/3361785.3361805>

commercial solutions demands a contact with the supplier that will study the needs and will implement some kind of customized version of the product. This is uncomfortable because it is not obvious how well the chatbot will perform and the doubts are more than the answers.

At this point, our feeling is that the most balanced solution for an enterprise with some development skills is to use available technologies and develop its own solution.

The above exposition justifies the need for a roadmap for those who want to develop a virtual assistant using a framework or a pre-built solution.

So this paper main goal is to build a framework that can help whoever needs to build a chatbot, to navigate through the available technologies towards an adequate solution for himself.

Consequently, this paper starts with an overview of what a chatbot is, to ensure a clear idea of the main subject in discussion. Afterwards, it presents the result of a literature review about chatbot technologies. Based on that knowledge one builds a taxonomy to compare the several available technologies. Finally, this taxonomy is used to find the best solution for a real problem in a large enterprise where the implementation of a chatbot was required

The proceedings are the records of the conference. ACM hopes to give these conference by-products a single, high-quality appearance. To do this, we ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download a template from [2], and replace the content with your own material.

2. CHATBOTS

To understand what type of technology is needed, the first thing to do is to understand what a chatbot is, and what scientific literature says about it. Therefore, this topic is all about chatbot related definitions as well as some history about it.

One definition is that a chatbot is a computer programme which interacts through natural language with a human user. At the beginning, chatbots were only applied to academic research but with the high development of this technology, it has the main objective to facilitate different sectors as sales, educational, social and technical support.[1]

Another definition is that a chatbot is a system that tries to perform a dialog, exchanging messages, always with the main purpose of mimic natural communication. Many domains already use it on the first interaction with a customer or technical support. However, at this moment, it is more and more used on education, mainly on distance studies and learning systems.[2]

The need to develop chatbots starts to increase with the vulgarization of personal machines and the desire to communicate, allied to the providers' innovation desire. Since humans use natural language to communicate, they wish to use that language to interact with their own personal computer. It is established that the best way to simplify the interactions between human and machine is to allow humans to express their desires or requests naturally by writing or speaking. To aim this goal, countless chatbots were developed, and they became known as software that interacts using natural language. Along the years, many chatbots were implemented, being a reference to the developers. There are some examples as ELIZA(1966), MegaHAL(1996), CONVERSE(1999), ELIZABETH(2002) HEXBOT(2004), among others.[3]

At this moment, chatbots have lots of attention on the business community due to the possibility to reduce resources on call centres and with the capability of dealing with multiple clients at the same time. The implementation of chatbots requires a correct analysis of its future use, aiming to prepare them to give the appropriate answer in each scenario. In many scenarios, the chatbot has not all the information to give a proper answer. In these cases it needs to be retrained, so it can reproduce a more natural interaction.[4]

In 2016, the most widespread technology was the chatbot. We have already listened to different names such as virtual assistant, conversational interface or just chatbot. The basic concept is the same: using natural language to communicate with a machine to accomplish a purpose. Increasingly visible on technology vanguard are the Big Four digital assistants Siri from Apple, Cortana from Microsoft, Alexa from Amazon and Assistant from Google. Following their lead, there are thousands of chatbots based on the text conversation. Many people see this technology as a revolution in interactions between human and machines. The MIT Technology Review lists the conversational interfaces as one of the ten most innovative of 2016. In January of 2016, Chris Messina from Uber writes an article declaring 2016 as the year of commercial conversation. In Mars, the CEO of Microsoft, Satya Nadella, announces that chatbots will be the next big technology, at the same level as graphic user interface, web navigator and touch screen. In April, Mark Zuckerberg proclaims that chatbots will be the solution to solve the overload of applications. Ignoring the normal controversy about public statements, the underlying premise is undisputed: the interaction with technology by text or voice, always in natural language, will be increasingly viable and potentially very significant.[5]

Creativity is one of the most fundamental characteristics of human intelligence and a big challenge for artificial intelligence. Techniques can be used to create new ideas in three ways: producing new combinations of common ideas, exploring the potential concept space and by transforming the idea creation. Artificial intelligence will have less difficulty in modulating the generation of ideas than the automation the evaluation.[6]

Currently, research into artificial intelligence and chatbots is increasingly going beyond limits that we thought would take longer to be achieved. In addition, areas such as Natural Language Understanding or Natural Language Processing are being explored to help the creation of more evolved chatbots. For some researchers, this exploration has passed ethical boundaries, such as trying, that robots learn to use feelings to convey a certain intelligence. For instance, they can serve as a companion to a human being that does not have the possibility to take care of an animal, either because of time or space restrictions.[7]

Natural language processing is a topic that is widely discussed and researched in these days. As it is one of the oldest research areas of machine learning, it is used in important fields such as voice recognition, machine translation, and word processing. Natural language processing has taken a breakthrough in the area of computing and artificial intelligence. The reason natural language processing is so important for the future is to help build models and processes that pick up pieces of information in the form of voice, text, or both by manipulating them as part of an algorithm inside the computer.[8]

Research on combining analysis programmes and natural language processing, which has recently begun to increase its attraction, is in an early stage of development. The existing work, although there are new comments, are mostly neglected because they are used for communication between developers.[9]

Natural language understanding is an area of natural language processing, which involves semantic and pragmatic knowledge, making it challenging because of the ambiguity of language and subtlety of the different perceptions that human beings have for different words. Natural language understanding attempts to achieve the meaning of communication language by giving the computer the ability to read and interpret the text. The key question, then, is how to derive the meaning of natural language, overcoming its inherent complexities. The natural language understanding, in turn, should allow one of the oldest objectives of the Artificial Intelligence that is the machine having the ability to read. Moreover, what would a machine learn as a result of this read, and what would it understand about the text and what semantic representation would it store? Well, new scenarios open up in which the machine has become empowered to analyse, aggregate, and reason huge amounts of information. It performs tasks that would be impossible for humans simply because of the scale and time involved.[10]

Interactive systems between humans and computers are growing thanks to their enormous potential and their great commercial value, which makes it even more tempting. With the development of neural models, the construction of a system of intelligent dialogue as our personal assistant or conversation companion is no longer a distant fantasy. Among most technologies the natural language understanding has a great challenge and continues to be extremely challenging, requiring that the system understands the context of the conversation and responds in an informative and coincidental manner.[11]

3. CHATBOT TECHNOLOGIES

In order to achieve a virtual assistant solution, the first prospection focused on ready-made solutions. However, the only commercial solutions found did not have very clear characteristics and required close contact with the supplier, who would then develop or adapt his solution to the concrete case that might be wanted. Commercial solutions like Answer Bot from Zendesk or the bot from perfect chat were discarded because they do not allow transferring the chat from the bot to a human agent when the bot exhausts the possibilities of providing an adequate answer. This feature may seem just one characteristic among others but in fact the absence of this possibility can lead to unhappy giving up clients which always leads to unpredictable business losses.

Therefore, a survey was done about existing technologies that make it possible to build a chatbot. I.e., architectural components that are available for those who may want to build their own virtual assistant.

The search focused on what the market has to offer. Components without a trial and enough documentation were discarded. From this search, six technologies proved to be adequate to use in a chatbot architecture and are described below in this topic.

3.1. Microsoft Bot Framework

As the name of the technology tells us, this is a framework powered by Microsoft which gives the possibility to create bots in many different ways. One of the greatest advantages is that it can be directly integrated with many other different technologies such as QnAMaker.ai or LUIS.ai also described in this topic. This technology allows using the Microsoft channels which allow the direct integration, for example, with Microsoft chat applications and with Facebook Messenger. Another relevant characteristic is that this framework works with code development, which means that not anyone can make a modification; every alteration needs to be developed. One big disadvantage is the weak connection with help desk live chat support.

3.2. QnAMaker.ai

This tool is also powered by Microsoft. This tool is highly used to integrate with Microsoft Bot Framework. It is a knowledge base designed to hold question and answer pairs. These pairs can support unlimited questions related just to one answer in order to match the users' possible different ways of writing. This tool provides a score in percentage of the matching between the question written and the question in the knowledge base. This score allows the developer to foresee different reactions of the chatbot based on different writing ways. This is an important advantage either when using it with Microsoft Bot Framework or when integrating it with other tools. Another great advantage of this tool is the possibility of using machine learning which allows that the most common questions not available in the knowledge base to be suggested for addition.

3.3. LUIS.ai

LUIS.ai is another Microsoft tool that allows very simple integration with the Microsoft Bot Framework. This tool is responsible for giving context to the conversation. That means that this is a key tool to build a chatbot that does more than answer questions and transfer the chat to human agents. This is a powerful tool for a bot integrated with many APIs and using many resources at the same time. Like the Microsoft tool approached in the previous point, it has an embedded system that allows knowing the score of the matching between the written and the several potential contexts. It has the capability to train itself with machine learning like the previous one.

3.4. NodeRed

This is a technology open source powered by IBM. It allows using many different flows, many of them developed by the community of NodeRed developers. This has the advantage of the large extensibility of modules, which means that it is connectable with many APIs. The main disadvantage is that the trust in this technology is on its community, which means that the support of the community can be weak and can stop the investment at any time. Another big disadvantage is that it has no connection with a help desk live chat support and one has to develop a channel for this purpose.

3.5. Recast.ai

This is a web platform powered by SAP which allows easily creating a bot to perform actions or to interact as a chatbot. This is a flow development meaning that it can be reprogrammable with

basic changes. The person who makes this alteration doesn't need high knowledge about any development language. This can be a disadvantage as well because anyone who has access to the account where the development has been done can easily modify all the bot architecture. Another disadvantage is that the simplicity of the created bot can limit the user interactions possibilities. In this technology to add a new pair of question and answer a lot of changes and reprogramming is needed just to cope with that simple addition. Another disadvantage is the weak connection with a help desk live chat support which means the necessity of finding another way to build a connection. It has the advantage of being improved by humans with suggestions given by machine learning.

3.6. DialogFlow

This platform is identical to the previous one, but powered by Google. It is a web platform that allows creating flows. These flows can interact with other technologies. This is accomplished in a way that allows easily creating a bot. Like in the previous solution, this simplicity leads to be suspicious about the possibility that anyone with access can do several alterations to the chatbot operation. The greatest disadvantage is the same as the previous one; a small addition means a lot of flow development. DialogFlow allows developing machine learning so the flows can be trained. This manages to be a very useful advantage because this way nobody needs to look at every chat done in order to find the keywords needed to train the model.

4. CHATBOT TECHNOLOGY CHARACTERISTICS

From the former analysis to the existing chatbot technologies, one elicited a set of relevant characteristics that should be considered when choosing architectural elements to build a chatbot solution. And those are:

- Flow Development;
- Natural Language Processing (NLP);
- Natural Language Understanding (NLU);
- External application integration;
- Open Source;
- Help Desk CRM Integration;
- Simple data insertion;
- Conversational Contexts;
- Question and Answer (Q&A) Knowledge Base;
- Conversational Memory;
- Machine Learning.

4.1. Flow Development

Flow Development is, like the name tells, developing software through the definition of flows between boxes of code. In these boxes the code is written and the boxes will be linked one to another until the flow, in this case to a conversation, is complete. This type of development can bring a huge advantage. That is, if for some reason the programmer needs to change the flow, he does not have to rewrite the code and only has to move a box to another place. On the other hand, this feature can also be a serious disadvantage because if a person with low programming skills has access to a flow, he can change it as easily as moving a box. If it were a code development that person would not dare to try to recode the flow. Therefore, this type of implementation must be always well weighted.

4.2. NLP and NLU

Natural language processing and natural language understanding, are often the core of a good implementation in any chatbot technology. Even though all the technologies have this functionality, it was left on the table to stress the importance of these characteristics. Nowadays is a mandatory feature.

4.3. External Application Integration

These technologies need to be integrated with other components of the system architecture. Some (or all of them) from other suppliers. Consequently, it is fundamental to know what kind of external application integration is provided. This feature allows connecting the chatbot to the real world, to an unlimited set of information. For instance, APIs to know the weather. It is important that the bot can create a support ticket or can send an email to the responsible for the company in charge for the bot, so he know what kind of trouble clients are passing through.

4.4. Open Source

Open source products have pros and cons and therefore this characteristic must be weighted. They are usually free, so on the short term, they are less expensive. If there is a significant community contributing to his improvement that is also a pro. Furthermore, there are fewer restrictions on what you can do with it. On the other hand, if the product does not fit some of your needs you can end up having to do it yourself. Open source products usually take more effort to master and to shape to your needs. Moreover, no one feels responsible for helping you or ensuring that the project will not be abandoned.

4.5. Help Desk CRM Integration

Integration with help desk customer relationship management (CRM) is the key feature to build a really helpful virtual assistant. A help desk CRM is a system that provides a web chat to allow customers interacting with the support team through text. This feature is probably the most important of these features. All these technologies claim to integrate with it. However, in the tests realized, four of the six technologies studied, ended up failing, mainly because of time-out issues.

The integration with the help desk CRM can be indispensable if the system aims to change between the bot and human assistants within the same user interface.

4.6. Simple Data Insertion

The issues related with data insertion are crucial because the ultimate success of the chatbot depends on the existing data. Data insertion needs to be a simple process with no need of redeveloping every time a question is need to be added or changed in the knowledge base.

4.7. Conversational Contexts

The bigger the knowledge base is, the more difficult it becomes to understand what is the user talking about. The virtual assistant supports several areas. Therefore, the knowledge base must have subareas, named contexts. That prevents the system to answer to a question with information from another context. The tool can then head the user to the right context and consequently to the right knowledge base.

4.8. Q&A Knowledge Base

Questions and answers knowledge base is the storage for the questions and answers pairs. This will be consumed by the chatbot to give the answers to the users. This feature is important because if there is not a local knowledge base owned by the enterprise, it is

necessary to consume an external knowledge base which means more costs to the company.

4.9. Conversational Memory

Conversational memory is to provide the bot with a memory that can remember all the conversation. This allows the bot to deal with situations like for instance: a customer asks the bot to give him the status of a ticket; next the bot asks him for the ticket number; instead of giving the ticket number, the customer asks for something else and after a while he finally gives the number; then, the chatbot remembers that the number is from the ticket. In other words, this allows the user to change contexts and the chatbot can still remember what was the first question that the user asked.

Conversational memory enables the chatbot with the capability of not losing context. Keeping track of past questions can help to understand the last question.

4.10. Machine Learning

Machine learning is the wording used to express the ability to elicit knowledge such as what are the most usual questions that users ask to the virtual assistant. This is a feature of great value for the knowledge base maintenance and in chatbot improvement.

5. COMPONENTS CONFRONTATION

Crossing the chatbot technologies with the characteristics meanwhile identified, one can build a table that eases the comparative analysis of these characteristics. See Table 1.

Table 1 Technologies comparison

Features	Technologies					
	<i>Recast.ai</i>	<i>DialogFlow</i>	<i>NodeRed</i>	<i>QnA Maker.ai</i>	<i>LUIS.ai</i>	<i>Microsoft Bot Framework</i>
Flow Development	X	X	X			
NLU and NLP	X	X	X	X	X	X
External Application Integration	APIs	All	All	APIs	APIs	All
Open Source			X			X
Help Desk CRM Integration				X	X	
Simple Data Insertion	X	X			X	
Conversational Context	X	X			X	
Q&A Knowledge Base				X		
Conversational Memory						X
Machine Learning	X	X		X	X	

With the above table, it becomes easier to see the features that are or are not present in each technology. As not all the features are present in all the technologies, being aware of the intended features for a specific implementation, results in an automatically narrowed choice.

Concerning flow development, three of the analysed technologies have this functionality, which represents one step ahead in bot development. These three are the Recast.ai, DialogFlow, and NodeRed.

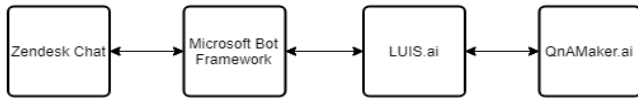


Fig. 1. Idealized Architecture

There are two features that none of these technologies miss. Those are NLU and NLP, and integration with external applications. Some of these applications can only integrate through APIs but others can integrate with all types of services such as an ERP or an external website information. Often to achieve all of the chatbot desired functionalities it is necessary to integrate the chatbot with several applications.

Only two of these technologies are open source, Microsoft Bot Framework and NodeRed. All the others are proprietary.

Integration with help desk customer relationship management (CRM) is probably the most important of these features. All these technologies claim to integrate with it. However, in the tests performed, only QnAMaker.ai and LUIS.ai have proved to deliver. All the other four technologies studied, ended up failing, mainly because of time-out issues.

Data insertion is simple in four of these six technologies. Microsoft Bot Framework and NodeRed demand the use of external applications to do data insertions.

The conversational context is native in half of the technologies. The other half consumes it from another application. This context has to be acquired from the conversation but applications such as LUIS can be integrated to give this context on the other applications.

The only native Q&A knowledge base is provided by QnAMaker.ai. All the others need to integrate an external one or to build it. Since we are searching for a tool that can answer to questions the other applications will need to use an external Q&A knowledge base to be able to store the multiple questions and answer pair.

The conversational memory is only native on Microsoft Bot Framework. For all the other technologies, this feature needs to be developed.

Machine learning is provided in most of these technologies. Even the two that do not have it can benefit of using the machine learning feature from another component of the system.

In general, the fact that some wanted specific characteristic is missing in one technology, is not reason enough to reject it. The architecture can provide that feature through another component.

6. CASE STUDY

This topic describes a case of applying the findings of this study to a large company in Portugal. That company wanted to improve the way they support their clients and to reduce operational costs of the help desk services. Their help desk CRM, already uses Zendesk Chat to support clients with human agents and they want

to maintain the same user interface. Therefore, an important restriction was the requirement of using Zendesk Chat. Keeping in mind the objective of reducing costs and improving response time, another important concern was that the client does not feel he is wasting his time. The main strategy was to broadcast simple doubts with a chatbot, replying them and forward to the human agent only the not so obvious answers. In order to get a complete virtual assistant with the maximum of functionalities, table 1 was used to select the adequate set of components. As the main objective of this project was to design a chatbot with a simple edition of the knowledge base of questions and answers we had to find a solution for that. The solution found was QnAMaker.ai as it has the best features and the best quality price relationship. Now one needs to select one or more components to connect the user interface supported by Zendesk, with QnAMaker. We decided to use our Help Desk CRM connected to the Microsoft Bot Framework with the LUIS.ai to provide the context of conversations and then the questions' context passing it to the QnAMaker.ai. This idealized architecture is pictured on Fig 1.

After analysing the above architecture, there are some important points to consider. First, Zendesk Chat provides an API in GraphQL which allows controlling how one passes the data to the chat and how one controls the conversations. The next point to consider is that all the intelligence from the bot is supplied by the Microsoft Framework. This framework uses LUIS.ai to obtain the conversation context and QnAMaker.ai to have a q&a knowledge base. This chatbot was built and connected to the help desk CRM but due to the weak integration between Microsoft Bot Framework and the GraphQL API, the approach had to been changed. The above architecture proved to fail due to connection time-out errors and duplicated messages. Therefore, a not so complete solution was adopted that matches all the fundamental requirements set by the company. These fundamental requirements are:

- Full compatibility with Zendesk Chat; this is important because the team that supports all the process uses this chat to communicate with costumers;
- To have a Q&A Knowledge base with easy data insertion,
- Have NLU and NLP so it can have some intelligence.
- And the optional requirement of having machine learning.

With these requirements and the purpose to improve the efficiency of the chatbot it was needed to understand what was the best architecture to solve the problems provoked by the previous integration. The solution found was to remove the Microsoft Bot Framework because the weak connection was only provoked by it. Furthermore, LUIS was also removed because LUIS is not actually needed once in this case study there is only one context of conversation. This approach implicates the development of a component between QnAMaker and Zendesk Chat. Therefore, all the intelligence needs to be provided by coding. This component was called Chatbot Brain in Fig. 2. This is the piece where all the conversation intelligence is.

In order to implement the pictured architecture, Chatbot Brain was then developed and proved to work perfectly. This chatbot solution has an architecture that it is not very common. On one side is the frontend, Zendesk Chat; on the other side is the knowledge base, QnAMaker; in the middle the connection between them supports all the interactions of the chatbot. This component is fully developed in javascript and has configured all

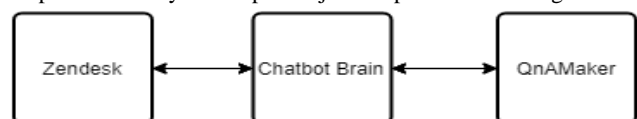


Figure 2. Final Architecture

the functionalities that Microsoft Bot Framework has but with the difference that it is all developed in code without using any framework. The response rate is instant and with this architecture, all the components do their job and they are all working properly.

Caution is mandatory when dealing with clients, so in this first trial, it was decided that the chatbot would never try to answer more than twice, before forwarding the client to the call centre human operator.

This solution is on production for about two weeks now and since the release date there is already some feedback from the costumers. A few costumers found the answer to the question they have asked; others felt impatient because they have been answered by a machine. The majority of the costumers did not express an opinion about their user experience because they have been transferred successfully to the human agent because the question was not on the knowledge base. This happened because in this first implementation we have only inserted a subset of questions for testing purpose. The customer satisfaction rate provided by Zendesk Chat back office has decreased ten percent, which was expectable and considered acceptable.

7. CONCLUSION

This paper consists of four main sections: What is a chatbot, existing chatbot technologies, its comparison and a case study.

In the first section, it is important to understand that there are many definitions of chatbot. Allied to this fact some articles have been written in the last few years. This lead us to conclude that this technology is on hype. The chatbots are improving every day so one can anticipate that many papers in this area will be written in the next years.

The second section, provides an overall approach of the six technologies studied. In this study, the technologies were tested by making prototypes of a chatbot integration between the help desk CRM and the set of technologies under study. This attempts in some cases failed due to incompatibilities described in that section. One big conclusion that can be taken by this is that the technologies are not in the expected state. Before this research, the expectation was that those chatbot technologies were more mature. This leads us to think that in the next years a big evolution will take place in this area.

The third section, the one that named this paper, relates the several technologies in order to facilitate the construction of the most adequate architecture for each application case. By analysing the table it is obvious that to achieve a complete solution one needs more than one technology. All of these technologies can be integrated together so that all these features can be achieved. However, as referred in that chapter they still need to be connected to a help desk. Therefore, in the current the state of the art of these technologies, it is still not possible to develop a complete solution only with these components.

Finally, there is a section describing the case study of an implementation done using the taxonomy developed. It describes the architectures that were tried as well as the difficulties and solutions found. Some detail was considered useful to explain how to reason about building an architecture and to overcome the limitations of the current technology. This case study is already implemented and being used in a large company.

Although functional, the developed solution does not use all the features exposed. While successfully tested in a controlled environment, the success for the future cannot be taken for granted. Only after more tests of this application in the real world, one can conclude about its stability and the types of client to whom the chatbot can be a success.

The case study was important to show the contribution of this work to the current knowledge about building chatbot solutions.

There are still some improvements that can be made to this implementation. There should be a feature to filter chitchat so that the bot does not process it as a technical question. Another improvement would be to automatically measure and analyse the responses of the bot with the purpose of collecting data to tune and improve it.

8. ACKNOWLEDGMENTS

This work has been supported by national funds through FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.

9. REFERENCES

- [1] Aquino, V. H. O. and Adaniya, M. H. A. C., 2018, *Desenvolvimento e aplicações de Chatbot*. Revista Terra & Cultura: Cadernos de Ensino e Pesquisa, 34(especial).
- [2] Paz, F. J., Silveira, C., Krassmann, A., and Tarouco, L. M. R., 2017, *Perspectivas tecnológicas para o aprimoramento de chatbots educacionais em AIML*, Rev. Iberoam. Educ. en Tecnol. y Tecnol. en Educ., vol. 20, pp. 7–15.
- [3] Shawar, B. A. and Atwell, E., 2007, *Chatbots: are they really useful?*, Zeitschrift für Comput. und Sprachtechnologie, vol. 22, no. 1, p. 29.
- [4] Reshmi, S. and Balakrishnan, K., 2016, *Implementation of an inquisitive chatbot for database supported knowledge bases*, Sadhana - Acad. Proc. Eng. Sci..
- [5] Dale, R., 2016, *The return of the chatbots*, Nat. Lang. Eng., vol. 22, no. 5, pp. 811–817.
- [6] Boden, M. A., 1998, *Creativity and artificial intelligence*, Artif. Intell., vol. 103, no. 1–2, pp. 347–356, Aug. 1998.
- [7] Boden, M. et al., 2017, *Principles of robotics: regulating robots in the real world*, Conn. Sci., vol. 29, no. 2, pp. 124–129.
- [8] Jain, A., Kulkarni, G., and Shah, V., 2018, *Natural Language Processing*, 2018. International Journal of Computer Sciences and Engineering, Vol.6, Issue.1, pp.161-167, 2018.
- [9] Nie, P., Li, J., Khurshid, S., Mooney, R., and Gligoric, M., 2018, *Natural Language Processing and Program Analysis for Supporting Todo Comments as Software Evolves*, Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI Publications.
- [10] Navigli, R., 2018, *Natural language understanding: Instructions for (present and future) use*, IJCAI Int. Jt. Conf. Artif. Intell., vol. 2018-July, pp. 5697–5702.
- [11] Zhang, Z., Li, J., Zhu, P., Zhao, H. and Liu, G., 2018. *Modeling multi-turn conversation with deep utterance aggregation*. arXiv preprint arXiv:1806.09102.