From Email-based to Chatbot-based IT Support: A Conceptual Design

[1]Lossan Bonde

[1]Assistant Professor
[1]Department of Applied Sciences,
[1]Adventist University of Africa, Nairobi, Kenya.

Abstract: *Email-based IT support refers to IT support systems whereby an email address is set to receive requests from users. Upon receipt of an email at that address, a ticketing system triggers an incident ticket generation. Then the ticket is assigned to an IT support person who processes it manually and responds to the user. This process is subject to long delays since the assignment and processing require human intervention. The manual operation is not only time-consuming but also leads to a lot of workloads to IT support personnel and produces frustrations on the users' side. As chatbots technology gets more and more reliable, IT support is one of the areas where their use is much needed and appreciated. This research proposes to elaborate on a conceptual design of a framework for the production of chatbots for IT support. The solution operates in three steps: (1) mails in the mailbox are read, analysed and incorporated into a knowledge base (KB). (2) A machine learning algorithm (MLA) is applied to enrich it and make it ready for natural language processing. (3) The last step of the process produces the chatbot. The design and implementation of the proposed framework is a heavy project that is just starting. The scope of this paper is about the overall conceptual design of the framework. Our future work will address the details and implementations of the components.*

IndexTerms *- Chatbot, machine learning, artificial intelligence, ontology, natural language processing.*

I. INTRODUCTION

Chatbots have existed shortly after the 1950s when Alan Turing published his article for testing how a computer programme can mimic human brain. The first chatbot ELIZA was made in 1966, followed by PARRY in 1972, and many others down to ALEXA in 2014. Today, chatbots have moved from trying to imitate human to solving real-life problems. The advances in computer technologies (memory, CPUs and GPUs, etc), the progress in artificial intelligence in general and natural language processing in particular, are being laveraged to produce powerfull intelligent computer applications in virtually almost all areas of human activities.

In IT support, most of the user's requests (80%+) concern simple and recurrent matters for which automatic answers are possible. According to [12], in 2018, the most frequent request for IT support was the request for a password reset. Many other matters recurrently take the precious time and energy of IT agents to address them. This paper provides a solution for the production of chatbots which will take into consideration the history of support requests and be the upfront agent to deal with users requests, and then channel them to a human agent whenever the matter being too requires human intervention.

The size of this research project requires to break it into multiple milestones. This paper, the first milestone, aims at presenting the overall conceptual design of the proposed framework. The approach used recognises and takes into consideration that most current IT support services (help desks) are email-based. An email address or set of email addresses are provided for users to send their requests. An IT support personnel reads the mails and provides the solutions to the users by mails. Over the years, these emails pile up and form an important mine of data/information. Unfortunately, this valuable information does not contribute to the system. The proposed solution seeks to use this information as the basis for automatic answers. The approach is a three-step process: (1) mails in the mailbox are read, analysed and incorporated into a knowledge base (KB). (2) A machine learning algorithm (MLA) is applied to enrich the knowledge and make it ready for natural language processing. (3) The last step is the production of the chatbot.

The solution, as outlined above, starts by building a knowledge base (KB). If the existing IT support users' system has a knowledge base, the reading and parsing of the emails will update that existing KB. The KB, central in this approach, is designed based on an ontology of the IT support domain. Sections 3.2 and 3.3 introduce the definition of the ontology and the design of the KB system. The work of Jorge Freitas and al. in [5] is the basis of the ontology.

The next important component of the solution is the machine learning algorithm (MLA). The algorithm is at the heart of automation of responses to users. It is first used to enrich the KB and secondly, it contributes to the process of generating the code of the chatbot. The algorithm is complex and combines many ML techniques including supervised and unsupervised learning, natural language processing and deep learning. Section 3.4 gives the details of the algorithm.

Finally, the last step in the process is the code generation of the chatbot. This step involves many technologies, including Model-Driven Engineering, natural language processing, knowledge discovery, etc. Section 3.5 gives an overview of the code generation system and how various technologies are put together to produce the chatbot.

Though the research has made a significant contribution in the area of the use of chatbots in IT support, a major limitation is that the solution only considers support requests originated from emails. Whereas in IT support, many other sources are used (phones, web portals, forums, and chats).

The immediate future related research work will revolve around the implementation of the proposed framework. Once the framework is implemented and evaluated, subsequent objectives will include (1) the support of other sources of IT users' support requests like phones, forums, chats, etc. (2) extend the solution to integrate other IT services as the current proposal is limited to IT support.

The remaining part of the paper is structured as follows: section 2 gives state of the art in using chatbots for IT support. Relevant research related works are presented and commented. Section 3 is the heart of this paper. It introduces the proposed framework and provides all details necessary to understand the contribution. Finally, section 4 concludes the paper and presents the perspectives of this work.

II. CHATBOT DESIGN AND IMPLEMENTATION

As Aditya et al. in [1] rightly state, chatbots represent an important move in human-computer interactions. The applications of chatbots range from financial advisors, legal aid, preliminary medical advisors, to e-commerce. The great contribution of chatbots lies in automation, which significantly reduces human efforts.

In [2], Shawar and Atwell have made an effort to answer the question: "Are chatbots really useful?" They presented real-life applications that involve chatbots. Though IT support is not in the list of examples of applications, the ChatFAQs as presented applies to IT support. Other applications in education and e-commerce also apply directly to the IT support sector.

As chatbots become more and more popular, their application in IT support - especially in support service desk - is of great interest. In this section, we present the state of the art of chatbots in IT support management systems.

Zevik Farkash [16] gives a summary of issues and concerns in the IT support that can be addressed by the use of chatbots. The list of possibilities is long, but the most important ones are:

- **Providing technical support to mobile-first consumers**: Mobile devices are becoming the primary devices that most people use for both work and entertainment. Companies are now forced to align their provision of technical support to suit the situation. Chatbots programmed to resolve simple and repetitive IT support tasks is an efficient way of addressing this matter.
- **Maintaining acceptable levels of service**: As the author of [16] rightly put it, "realising the vision of a fully resourced help desk would require countless hours of recruiting, training and motivating the support staff, with the view to providing them with the right combination of hard and soft skills". The way is to develop online chatbots driven by the right scripts and actions that will offer and maintain a human alike level of service in an indefinitely time.
- **Optimising support processes**: another area where chatbots can be very efficient compared to humans is in recording histories of their conversations with every single customer. The quality control department can then analyse these historical records and advise on possible improvements aiming at optimising the support process.
- **Encouraging self-resolution**: Even though companies IT support department may provide FAQs, wikis and other resources to enable users to troubleshoot and resolve some of their issues without referring to IT agents, most of the time users will continue to struggle with self-service and keep returning to IT support. Chatbots as assistants to the users, and even the process of using self-service support will be much more effective.

From the above, it is obvious that chatbots are being explored and designed for IT support. In the remaining of this section, the focus is on how are those chatbots are designed and implemented. A study of the abundant literature on chatbot design and implementation enables to categorise the approaches and technologies into four classes: (1) Knowledge Base, (2) Information Retrieval Technique, (3) Natural Language Processing, (4) Programming Languages.

2.1 Knowledge Base

Knowledge base (KB) is an important component of most chatbots. The term knowledge base first appears with expert systems, which known as knowledge-based systems. According to Wikipedia[1], a knowledge base is a system that is used to store complex and unstructured information so that computer applications can process them. Approaches and technologies used in the design of chatbot knowledge include handcrafted systems [7], and other techniques based on the Artificial Intelligence Markup Language (AIML) and the Apache Unstructured Information Management Architecture (UIMA). [4], [10], and [1] describe these latter approaches.

The handcrafted systems, as described in [14, page 2], a knowledge engineer studies the literature of the domain, and in dialogue with experts of the domain, creates a computer-readable knowledge base. This approach is not suited for automatic generation, and therefore, is not considered for this study.

Several chatbot technologies make use of the AIML language to design their knowledge base. Among those technologies are: Mitsuku, ALICE [1], [8], CHARLIE [3], and [13].

Artificial Intelligence Markup Language (AIML) provides several functionalities, which makes it be a good candidate for chatbot knowledge base systems. The most important of those functionalities are:

- **Knowledge Unit**: AIML provides the tag <***category***> to represent a knowledge unit.
- **Pattern Matching**: AIML provides a pattern matching system which enables user input mapping with adequate responses of the chatbot. This matching system greatly simplifies both the programming of the chatbot and information retrieval.
- **Power of Srai Tag**: The <***srai***> tag provides many important features that are key to the operation of a chatbot: Symbolic Reduction, Divide and Conquer, Synonyms Resolution, and Keywords Detection.
  - *Symbolic Reduction*: AIML provides through the <***srai***> tag an elegant mechanism to reduce complex grammatical patterns to simple patterns. The symbolic reduction feature allows the chatbot to recognise human equivalent expressions.
  - *Divide and Conquer*: The <***srai***> tag is also helpful in reducing multiple categories and the reuse of sub sentences. The divide and conquer capability allows the construction of complex replies.
  - *Synonyms Resolution*: Synonyms are very important in conversations. The <***srai***> tag provides a mechanism to resolve synonyms. For example, by using this tag, a chatbot recognises that *factory* is a synonym to *industry*.
  - *Keywords Detection*: Finally, keyword detection enables the chatbot to return a simple response when a user types a specific keyword. For example, if a user enters the word *school* anywhere in a sentence, the chatbot will detect that keyword and return an appropriate response.
- **Context Handling**: The ability of a chatbot to respond based on context is an important feature. Fortunately, AIML has the following tags: <***what***>, <***topic***>, and <***condition***> that are used to achieve that feature.

All these features and capabilities have made the AIML language a good candidate for the design of the knowledge base in this study. Moreover, as rightly assessed by [11, pages 2-4], the AIML 2.0 specification [9] has resolved some of the shortcomings of AIML and introduced the support of access to external resources, knowledge acquisition, and IoT devices.

1 https://en.wikipedia.org/wiki/Knowledge_base

IBM DeepQA chatbot [1] uses Apache Unstructured Information Management Architecture (UIMA). Apache defines UIMA applications as software systems that analyse large volumes of unstructured information to discover knowledge that is relevant to an end-user[2]. A UIMA application might ingest plain text and identify entities, such as persons, places, organisations; or relations, such as *works-for* or *located-at*. UIMA is more than a language; it is a complete framework that enables to split your application into various components. Using UIMA to implement a knowledge base as a standalone component out of the framework and be used by other tools might be a challenging task. In this study, AIML shall be prefered over UIMA.

2.2 Information Retrieval Technique

The way the chatbot retrieve information from the knowledge base is a vital step in the design. From literature, there are mainly two classes of information retrieval methods: **Retrieval-based methods** and **Generation based methods**.

In **Retrieval-based methods** [15], the first step is to retrieve the most relevant Question/Answer pair from a set of existing Question/Answer pairs, which best matches current utterance Question based on semantic matching models, then returns Answer as the response. One disadvantage of such a method is that, for many specific domains, collecting such Question/Answer pairs is impracticable.

The **Generation based methods** use in general an encoder-decoder framework which first encodes the question Q as a vector representation, then feeds this representation to a decoder to generate the response A. Similar to retrieval-based methods, such approaches also depend on existing Q/A pairs as training data. Like other language generation tasks, such as machine translation and paraphrasing, the fluency and naturality of machine-generated text is another drawback.

To overcome these difficulties, Zhao Yant et al. [15] introduce a novel way to retrieve information for chatbot engines. In their approach named *DocChat*, to find responses, for each user utterance, instead of looking for the best Q/A pair or generating a word sequence based on language generation techniques, they select a sentence from given documents directly, by ranking all possible sentences based on features designed at different levels of granularity. Using documents rather than Q/A pairs is greatly preferred as it improves the adaptability of chatbot engines on different chatting topics.

2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) is indispensable to the construction of chatbots. A successful conversion between human and the chatbot requires a set of NLP techniques: (1) Natural Language Understanding (NLU) to transform the human sentences into a form of a structured ontology. NLU is used mainly for the transformation of the user input, (2) then Regular Language Generation (NLG) comes in to transform the machine-generated responses into the user language. Most chatbots implementation use the Natural Language Toolkit (NLTK) (a set of libraries) along with various programming languages.

2.4 Programming Language

Any programming language can serve to write the code of a chatbot. In the study of "A Survey of Various Chatbot Implementation Techniques" [1], Java, Javascript, Objective C, Python, and Node.js are the most used languages in the implementation of recent chatbots. Python is, by far a good choice because of the integration of various tools for Machine Learning (ML), Natural Language Processing (NLP) and Natural Language Tool Kit (NLTK).

III. PROPOSED FRAMEWORK

The proposed solution is to define a framework that enables an IT support system to move from email-based support to chatbot-based support. The following summarises the assumptions behind this work:

- **Email-based IT Support**: many IT support systems are email-based; users send their queries to an IT support email and wait for the IT personnel to read the emails and respond to the user. The old emails in the IT support mailbox are valuable information to consider, analyse and extract so that they are included in a knowledge base and then used as training data to a machine learning algorithm.
- **Simple and recurrent users requests**: Most users requests are about simple and recurrent issues. The solutions to those queries can be documented and stored in a knowledge base, and then retrieved automatically by a chatbot. This approach will not only release the IT personnel from attending to so many issues and concentrate on more important matters, but it will also provide instant solutions to users. Therefore, saving time for both IT support team and users.
- **Machine Learning**: a machine learning algorithm (MLA) can be used to provide answers to most users' requests. The MLA will also serve to generate new knowledge that goes back to the knowledge base.
- **Code Generation**: Implementing a chatbot is tedious and time-consuming work. Given the MLA and knowledge base, it is possible to generate a significant part of the chatbot code.

The idea behind this work is that chatbot can be used to efficiently assist IT support teams in delivering quick and reliable solutions to users inquiries.

2 https://uima.apache.org/

3.1 Overview of the Framework

Figure 1 gives an overview of the proposed solution.

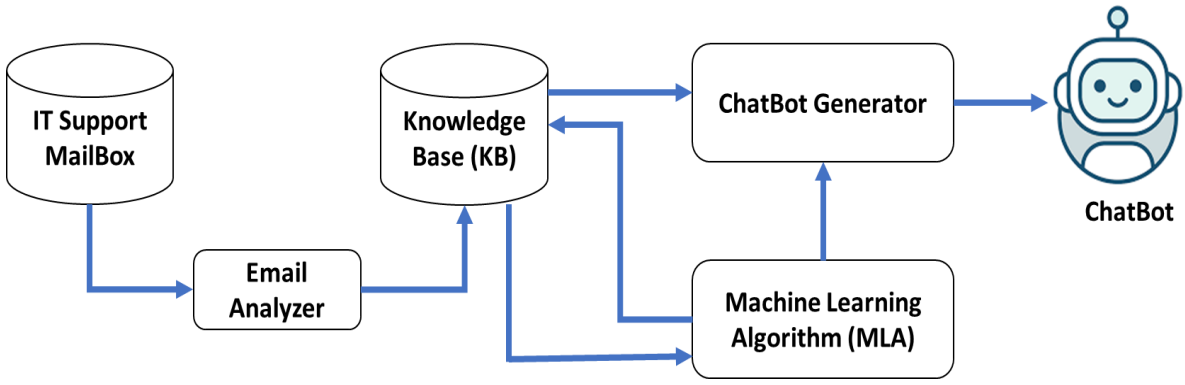

Figure 1 - Overview of the Framework

The solution contains four main components: (1) the ***Email Analyser*** (EA) which fetches old the emails in the mailbox and extract relevant information into the knowledge base. (2) the ***Knowledge Base*** (KB) is at the heart of the framework and stores relevant information for the conversations with the chatbot. (3) the ***Machine Learning Algorithm*** (MLA) reads the data in the knowledge base then performs knowledge discovery operations, and the newly created knowledge is stored back in the KB. (4) the ***ChatBot Generator*** (CG) uses the KB along with the MLA to produce the code of the chatbot.

3.2 Ontology for IT Support

Defining an ontology is fundamental in determining the nature and structure of the information to store in the knowledge base. In [5], Jorge Freitas et al. have developed a comprehensive ontology for IT services. That ontology contains six packages. The proposed solution selects relevant packages from their work and adapts them to fit the needs of the project. The details of the finally adopted ontology will be worked out separately in subsequent papers.

3.3 Knowledge Base System (KBS)

Knowledge-based systems are powerfull computer systems capable of making decisions and undertake actions based on the knowledge residing in them. They can understand the context of the data they process. These capabilities of KBS made them a very promising avenue for chatbots. A KBS has two components: The ***Knowledge Base,*** which is a kind of knowledge repository, and the ***Search Engine,*** which is the interface of the KBS. Another important aspect of KBS is its capability to learn from existing knowledge and produce new knowledge. The proposed solution implements the search function in the chatbot, and the learning aspect in the MLA.

Through the knowledge system, knowledge is represented explicitly through the use of ontology, as suggested in [6]. This approach does not only release the programmer from implicitly specifying knowledge through code as done in traditional programming, but also provide flexibility and scalability.

3.4 Machine Learning Algorithm

A machine learning algorithm is used for three purposes: (1) to refine the information in the knowledge base, (2) to generate the code of the chatbot, (3) to serve as a backend system to the chatbot search engine. Figure 2 represents the operational model of the chatbot. It shows how the MLA is at the centre of that operation.

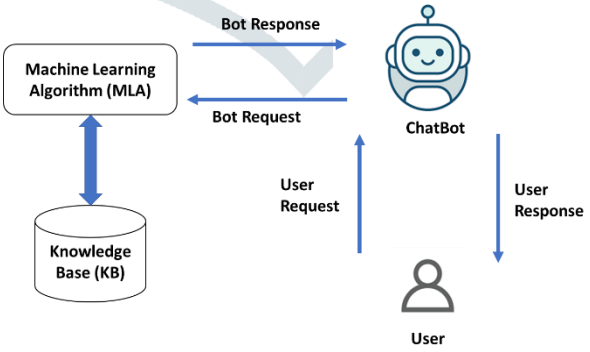

Figure 2 - Interaction between User, Chatbot and MLA

The framework design carefully reduces the responsibility of the chatbot so that the code generation is possible. The chatbot interacts with the user to receive requests which it pre-processes using Language Processing techniques and send standardised queries to the machine learning algorithm that will carry out the heavy processing duties.

3.5 Chatbot Code Generation

The chatbot design contains two components: a ***User Interface***, and a ***Transformation Engine***. The user interface is a web-based application which handles the conversation with users requesting IT support. The transformation engine is responsible for converting the user requests into a standard form that will be further processed by the machine learning algorithm.

- **User Interface:** The implementation of the user interface uses standard web development tools. Part of the code, including CSS files and configuration files, are hand-coded before the code generation. The actual code that handles the conversion with users will be javascript based (Node.js and angular).
- **Transformation Engine**: the engine code is Python-based. NLTK is intensively used to make the conversion of user requests and bot responses.

Subsequent phases of the project will determine the code generations tools and techniques.

IV. CONCLUSION AND FUTURE WORK

This paper has presented a conceptual design of a framework for IT support chatbot construction. The work enables email-based IT support systems to move into a chatbot-based system. The solution incorporates all previous emails received in the mailbox. A machine learning algorithm is used to provide automatic answers to users' queries. Thus releasing the IT support personnel from repetitive issues that tend to consume their valuable time on one hand. On the other hand, The chatbot solution is also prefered by users as it provides instant answers, and avoid frustrations of unattended requests.

The paper reports on the first phase (conceptual design) of an ongoing project; therefore, technical details are missing. Apart from the missing details, a major limitation of the work presented here is that the current framework does not support other means of contacting IT support (phone, chats and forums, etc.).

Therefore, the future work of this project will include (1) the implementation, testing and evaluation of the design, (2) the support of phone, forum, web portals and other means of contacting IT support, and (3) the extension of the framework to other IT services.

REFERENCES

[1] A. Deshpande, A. Shahane, D. Gadre, M. Deshpande, and P. M. Joshi. 2017. A survey of various chatbot implementation techniques. International Journal of Computer Science and Applications, XI(Special Issue).
[2] B. Shawar and E. Atwell. 2007. Chatbots: Are they really useful?. LDV Forum, 22.
[3] F. A. Mikic, J. C. Burguillo, D. A. Rodriguez, and E. Rodriguez. 2009. Charlie: An AIML-based chatterbot which works as an interface among machines and humans.
[4] J. Cahn. 2017. Chatbot: Architecture, design, & development.
[5] J. Freitas, A. Correia, and F. Brito. An ontology for IT services.
[6] L. V. Naykhanova, I. V. Nyakhanova. 2018. Conceptual Model of Knowledge Base System. International Conference Information Technologies in Business and Industry. IOP Conf. Series: Journal of Physics, 1015 (032097).
[7] M. J. Pereira and L. Coheur. 2013. Just.chat - a platform for processing information to be used in chatbots.
[8] P. V. Rosmalen, J. Eikelboom, E. Bloemers, K. V. Winzum, and P. Spring. 2012. Towards a game-chatbot: Extending the interaction in serious games.
[9] R. S. Wallace. 2014. AIML 2.0 Working Draft, https://docs.google.com/document/d/1wNT25hJRyupcG51aO89UcQEiG-_HkXRXusukADpFnDs4/pub
[10] S. A. Sheikh, S. Singhal, and V. Tiwari. 2019. Artificial intelligence-based chatbot for human resource: A survey. Journal of Emerging Technologies and Innovative Research, 6(1).
[11] S. Arsovski, A. D. Cheok, M. Idris, M. R. B. A. Raffur. 2018. Analysis of the Chatbot Open Source Languages AIML and ChatScript: A Review. Published through ResearchGate – https://www.researchgate.net/publication/323279398
[12] T. Burgess. 2018. Solar winds service desk.
[13] V. Bhargava and N. Maheshwari. 2009. An intelligent speech recognition system for the education system.
[14] X. Huang and J. R. Jensen. 1997. A machine-learning approach to automated knowledge-base building for remote sensing image analysis with GIS data.
[15] Z. Yant, N. Duan, J. Bao, P. Chen, and J. Zhou. 2016. DocChat: An information retrieval approach for chatbot engines using unstructured documents.
[16] Z. Farkash. 2018. IT support chatbot — Tech support challenges and what chatbots can do about them.