



**HAL**  
open science

# Rexy, A Configurable Application for Building Virtual Teaching Assistants

Luca Benedetto, Paolo Cremonesi

► **To cite this version:**

Luca Benedetto, Paolo Cremonesi. Rexy, A Configurable Application for Building Virtual Teaching Assistants. 17th IFIP Conference on Human-Computer Interaction (INTERACT), Sep 2019, Paphos, Cyprus. pp.233-241, 10.1007/978-3-030-29384-0\_15 . hal-02544593

**HAL Id: hal-02544593**

**<https://inria.hal.science/hal-02544593v1>**

Submitted on 16 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# *Rexy*, a configurable application for building Virtual Teaching Assistants

Luca Benedetto and Paolo Cremonesi

Politecnico di Milano, Milan, Italy  
{luca.benedetto,pao1o.cremonesi}@polimi.it

**Abstract.** In recent years, virtual assistants gained a pervasive role in many domains and education was not different from others. However, although some implementation of conversational agents for supporting students have already been presented, they were *ad hoc* systems, built for specific courses and impossible to generalize. Also, there is a lack of research about the effects that the development of systems capable of interacting with both the students and the professors would have. In this paper, we introduce *Rexy*, a configurable application that can be used to build virtual teaching assistants for diverse courses, and present the results of a user study carried out using it as a virtual teaching assistant for an on-site course held at Politecnico di Milano. The qualitative analysis of the usage that was made of the assistant and the results of a post study questionnaire the students were asked to fill showed that they see conversational agents as useful tools for helping them in their studies.

**Keywords:** Artificial Intelligence · Conversational Agents · Human-Computer Interaction · Virtual Teaching Assistants · NLP

## 1 Introduction

For several years research has been exploring the usage of conversational agents - also referred to as chatbots - in education, due to their capability of bringing advantages both to students and to teachers. Recent works, such as [3], suggested the use of chatbots as Virtual Teaching Assistants (VTA) to enable scalable teaching and reduce instructors' workload. Thanks to VTAs, students can have most of their requests answered immediately without having to wait for the teacher to write back; at the same time, instructors can focus on other aspects of the teaching activity that necessarily require a direct human interaction.

Regardless of these promising characteristics of conversational agents, there is a lack of studies to show the efficacy of VTAs. Previous works proposed chatbots as virtual teaching assistants, but they were either concepts without a working implementation or systems with limitations: in particular, even though VTAs are unavoidably able to understand only part of the requests they receive, the existing implementations can interact only with students and are not capable of asking human teaching assistants for help.

In this paper, we introduce *Rexy* - a configurable application that can be used to build VTAs about any types of courses - and present a user study in which *Rexy* was deployed as a virtual assistant for an on-site course held at Politecnico di Milano in order to understand how the students perceived the assistant, whether it was useful and how it could be improved.

The main contributions of this work consist in:

- the introduction of *Rexy*, a novel application to implement VTAs that can interact with both students and professors;
- the implementation and deployment of an assistant based on *Rexy*;
- a qualitative user study, which shows that students are willing to leverage VTAs and see conversational agents as useful supporting tools.

The rest of the paper is organized as follows. Section 2 presents the state of the art; Section 3 introduces *Rexy*; Section 4 provides the details of the user study, whose results are presented in Section 5. Lastly, Section 6 presents the final discussion and the directions for future works.

## 2 Related Work

The effectiveness and usefulness of instant messaging applications and forums to support online and on-site students has been proved by several works [14, 12, 13]. Unfortunately, there are many constraints that limit the supplying of continuous human support, such as the cost and availability of Teaching Assistants (TA); a possible solution consists in automating student’s support via the usage of conversational agents.

Conversational agents and their possible applications in diverse domains have been studied for years [10], and researchers already tried to use them in education several years ago [11, 9], creating the first implementations of what we call “virtual teaching assistants” (VTA). However, although some of these results are still relevant today, the technology available at the time did not allow the implementation of chat-based VTAs that were really effective in improving the students’ learning experience. More precisely, conversational agents rely on natural language processing (NLP) [4] techniques to understand the users’ requests and provide meaningful responses, and the algorithms available at the time were not accurate enough for the tasks that had to be solved in VTAs.

In recent years, thanks to the improvements in NLP, researchers returned to explore the usage of conversational systems in education, but there is still a lack of studies to show the efficacy of VTAs. In particular, no previous works explored the possibility of building a VTA capable of interacting with both students and human TAs, and most works did not focus on the perception that the students have of a VTA, even though this is an important aspect to consider and might affect the effectiveness of such systems [8].

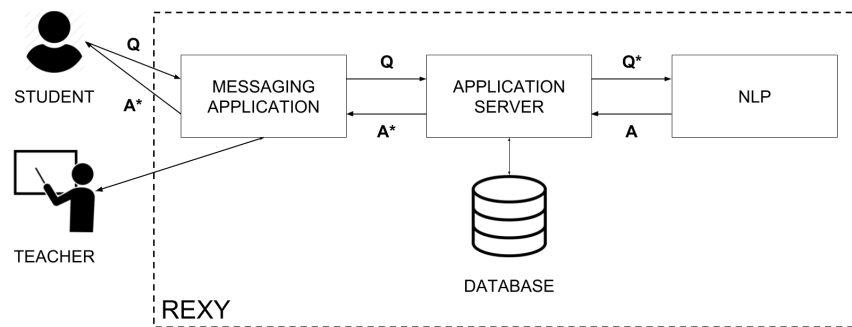
Goel et al. in [7] introduced Jill Watson (JW), a VTA for online education similar to the ones that can be implemented with *Rexy*; the authors showed the possible applications of JW in a MOOC, but they did not present the details of

the implementation and always considered it as a black-box. Also, *Rexy* leads to the development of VTA that have a different role: JW aimed at completely replacing human TAs, therefore it encountered some issues that still have to be dealt with [5]; on the other hand, our application is meant to work together with the professors (as suggested in [11]) and therefore does not have to deal with conversations that are out of the scope of the specific course for which it is deployed. This is an important difference between the two approaches: VTAs built upon *Rexy* are meant to cooperate with human TAs in order to answer the requests they cannot deal with on their own, and can do so by proactively starting a conversation with a TA and asking for help.

Ventura et al. [15] and Akcora et. al [1] proposed conversational agents as virtual teaching assistants, but their objective was different from ours: indeed, the systems presented in both the works can be defined as tutors whose role is guiding a student while he’s consuming the content of an online course, not being assistants that can answer general requests. Also, while Ventura et al. already presented preliminary results of the use of their assistant, the system proposed in [1] is only a concept which will require further work.

### 3 The Application: *Rexy*

We propose a configurable application that can be used to implement different VTAs, extending the work done in [2]. The only step required to build from *Rexy* an assistant for a specific course consists in creating the appropriate knowledge base and feeding the application with it. As shown in Figure 1 this application is composed of three components: (i) messaging application, (ii) application server, (iii) natural language processing component.



**Fig. 1.** Architecture of *Rexy*

### 3.1 Messaging application

The front-end of the system consists of a Slack application that enables students to interact with the VTA. Although several messaging applications feature the possibility of implementing conversational agents, we decided to use Slack since it allows both one-to-one and one-to-many interactions: indeed, a Slack workspace can be used in the educational domain as a hub where students can interact not only with the VTA, but also with each other and with the professors. Additionally, Slack has two interesting features: it allows conversational agents (i) to send interactive messages and dialogs, thus enriching the way in which students can communicate with the VTA (e.g. with multi-choice questions), and (ii) to begin the conversation with a student, therefore enabling proactive assistants that can send messages to students that are struggling.

### 3.2 Application server

The role of the application server consists in processing the requests  $Q$  coming from the students and managing how the NLP component interacts with the database in order to create the response message  $A^*$ . In particular, its tasks can be grouped into three categories.

- **Preprocessing:** this is necessary when the creation of the response requires some information which is not available in the NLP component. More precisely, the application server has to keep track of the context of the conversation, since the memory of the NLP component is limited to the current interaction with the student and, if he references something previously said, the server must take care of this and modify the input request before sending it to the NLP component. In Figure 1,  $Q$  is the original question while  $Q^*$  is the question after it has been preprocessed. Let’s consider this example:

STUDENT(Q) : How many inhabitants does London have?  
*Rexy* : As of 2019, it has 9,126,366 inhabitants.  
 STUDENT(Q) : How many inhabitants did **it** have in 1750?  
 STUDENT(Q\*) : How many inhabitants did **London** have in 1750?  
*Rexy* : In 1750, 740,000 persons were living in London.

In the second request, the student refers to “London” with the pronoun “it”; therefore, the application server has to modify the message in order to make the NLP component understand the request.

- **Post-processing:** analogously to preprocessing, if the response requires some information that is out of the reach of the NLP component, the application server has to modify the message before forwarding the answer to the messaging application. In Figure 1,  $A$  is the original answer while  $A^*$  is the question after it has been post-processed. An example is shown here:

STUDENT : How many inhabitants does London have?  
*Rexy*(A) : As of 2019, it has  $\langle num\_london\_inhabitants \rangle$  inhabitants.

*Rexy*(A\*) : As of 2019, it has 9,126,366 inhabitants.

The NLP component creates the template answer  $A$  and the application server fills the gap with the information gathered from the database.

- **Interaction with a human teaching assistant:** every time the NLP component receives a request forwarded by the application server, it generates a response and sends it, together with an estimated confidence level  $c$ , back to the server; a low value of  $c$  means that the NLP component is not confident in the generated response. A threshold  $t_L$  is defined in the application server; if  $c < t_L$  the assistant forwards via Slack the question to the human TA and sends to the student a notification stating that the request will be answered as soon as possible by a human. When the TA writes the response, this is sent to the student and the correct request-response pair is sent to the NLP component as well, in order to update the dataset used for future retraining.

The application server contains the information about the course, the information about the students and the history of all the conversations.

### 3.3 NLP component

The NLP component leverages IBM’s Watson Assistant [6], an AI engine offering services for natural language understanding and natural language processing; in order to make it work in the educational domain, we can train it with data consisting of a set of request-response pairs. Each request is labelled with an *intent* and/or an *entity*: *intents* represent the objectives of the users, while *entities* give a context to the interaction between the user and the assistant, thus affecting the way in which it reacts to each *intent*.

*Intents* are specific of the educational domain but do not depend on the particular course. Therefore, once we have defined the ones related to education, they can be used without modifications for every course; in particular, we defined 71 *intents*. Examples of intents we defined are *exam\_date*, which indicates that the student is interested in knowing the date of the exam; *content\_references*, when the student wants to know where to find a specific course topic in the material; and *course\_program*, meaning that the student wants to know the program of the course.

Differently from *intents*, *entities* are, in general, course dependent. Each *entity* does not represent a unique concept, but instead a set of concepts (named *values* in Watson).

Let’s consider the same example as above:

STUDENT : How many inhabitants does London have?

Here, “London” is a value of the entity *course\_topic*, “inhabitants” is a value of the entity *attribute* and the intent is *retrieve\_topic\_information*.

When implementing an assistant using this application, it is sufficient to enlarge the set of *entities* by inserting the ones that are specific to the course for which the VTA is being built. In our case, we configured and deployed *Rexy*

as a VTA to support the students enrolled in an introductory course about recommender systems held on-site at Politecnico di Milano. In order to adapt *Rexy* to this scenario, we only had to store in the database server the knowledge base containing the information specific to this course, and to define the course-specific entities. In total, we defined 231 values grouped in 4 entities.

A very interesting feature of Watson Assistant consists in the possibility of continuously retraining the model, thus enabling active learning. Our architecture leverages this in order to keep improving the accuracy of the model: every time a request is correctly answered (by the assistant or a human TA), the request-response pair is inserted in the training set so that it will be used for retraining. Thanks to this approach, the number of required interactions with the human TA is likely to continuously decrease as more training samples are being collected.

## 4 User Study

The course was attended by 107 students and they were informed about the possibility of using the VTA to answer their questions; however, they were not forced to use it and could still interact with human TAs for help. Moreover, in order to stimulate the use of the VTA, we let the students join the Slack workspace with arbitrary nicknames, therefore they could not be identified and associated with the final score of the exam.

Our qualitative study has two goals: (i) understanding the effectiveness of the assistant by analyzing the interaction logs, (ii) collecting feedback from the students in order to understand how they perceive conversational agents as VTAs.

At the end of the semester, the students were given a post-study questionnaire to capture their opinion about the assistant: the questionnaire comprised both closed-ended and open-ended questions. In particular, the following questions were asked:

- How would you evaluate the support provided by *Rexy*?
- Do you think virtual teaching assistants could help you, as a student?
- Which was the best feature of *Rexy*?
- Which was the biggest limitation of *Rexy*?
- How could *Rexy* be improved, in your opinion?

## 5 Evaluation

At the time of writing, 22 students have interacted with the assistant in 263 conversation turns, thus having an average of 11.95 messages per student.

### 5.1 System Usage

Not all the messages required both the identification of an *intent* and an *entity*: more precisely, 182 questions required *intent* classification and the remaining

81 both *intent* and *entity* classification. Manual inspection of the activity log showed that the VTA was always capable of correctly detecting the *entity*, while the accuracy on *intent* classification was about 70%: 127 out of 182 *intents* were correctly classified. Diving a bit more into the details, we noticed that out of the 55 requests that were not handled only 26 were questions that should have been answered, the remaining 29 were random messages or messages mocking the assistant. Therefore, the actual *intent* classification accuracy was 83%.

Most of the students' requests were related to two aspects of the course: (i) definitions and examples about the topics presented during the lectures and (ii) lecture and exam schedule. This suggests that VTAs can be particularly useful as helping tools for reviewing the course and quickly retrieve information about it; indeed, students can obtain the information they are looking for without having to search the course material or the course website for it.

The students' requests involved only 43 *intents* out of the 71 we had defined (60.56%). This is reasonable, since *intents* were arbitrarily chosen while designing *Rexy*, future work should focus on this aspect to understand whether a reorganization of them would bring some improvements to the effectiveness of *Rexy*.

## 5.2 Post-Study Questionnaire

The questionnaire was answered by 40 students, 11 of them interacted with *Rexy* at least once.

When asked whether they see VTAs as useful supporting tools for students, only 4 students claimed that they did not see any advantages coming from them. Interestingly, these 4 students had not tried *Rexy*, suggesting that the ones who tried to interact with the assistant foresaw the advantages that such system could bring to their learning experience.

When asked why they did not use the assistant, most of the students replied that the support received from the human TAs and the other students was usually sufficient. However, one answer to this question was particularly interesting: a student said that he did not use *Rexy* because he did not feel comfortable talking to a bot. This aspect should definitely be taken into consideration in future research in order to understand whether similar problems could be overcome, especially because in MOOCs the interaction with human TAs necessarily introduces long delays in the communication.

Only 3 students rated in a negative way the assistant; the others appreciated the possibility of receiving immediate answers, specifically for quickly finding online video lectures and information about the schedule and the topics presented in each lecture, which is in agreement with the usage we observed in the interaction log.

Although only few students rated in a negative way the assistant, we received some comments about the limitations of this first implementation of *Rexy*. In particular, some students criticized the fact that sometimes the assistant has to forward the requests to a human TA. At the same time, however, they appreciated the fact that they were not asked to resend the question because *Rexy* was



able to manage everything on its own and they only had to wait a bit longer before receiving the answer.

## 6 Conclusions and Future Work

In this paper we have introduced *Rexy*, a novel configurable architecture that can be used to build virtual teaching assistants. We have shown the implementation of a VTA built upon *Rexy* for an on-site course held at Politecnico di Milano and reported the results of a qualitative user study in which we analyzed the effectiveness of this application. *Rexy* received positive feedback from the students and proved capable of answering most of their requests. In doing so, it reduced both the workload on the Professor and the average time elapsed between sending a question and receiving the answer.

The proposed application leverages IBM's Watson Assistant for the majority of the tasks related to NLP and thus offers a fairly easy solution for implementing virtual teaching assistants. Indeed, the only effort required consists in the curation of the knowledge base related to the target course to feed the database with.

Feedback from students suggests that some improvement could be made: in particular, enlarging the training set would make *Rexy* more accurate in detecting the correct *intent* and therefore reduce the need of interventions from human TAs. Indeed, even though the students appreciated the fact that they were not requested to perform any additional request when *Rexy* could not understand a question, the delay introduced by waiting for a human teaching assistant to answer was the one element that was criticized by the students.

Future work should focus more on the choice of *intents* made while developing *Rexy*. Indeed, they were arbitrarily chosen while designing the application and, although they proved very effective once *Rexy* was deployed, it is possible that some slightly different configurations could work even better in recognizing the students' requests.

Another aspect to consider in future work is the value that is used as threshold in the application server to decide whether *Rexy* is confident in the generated answer. Indeed, it has a direct impact on the system's performance since it determines whether the assistant should contact a human teaching assistant for help and some improvements could be obtained by performing performance tuning to find the optimal value for the threshold.

The current architecture only deals with one-to-one interactions, communicating with students in private conversations; however, one of the reasons of choosing Slack as the front-end application consisted in the possibility of enabling one-to-many interactions and we are currently working on implementing such feature.

## References

1. Akcora, D.E., Belli, A., Berardi, M., Casola, S., Di Blas, N., Falletta, S., Faraotti, A., Lodi, L., Diaz, D.N., Paolini, P., et al.: Conversational support for education. In: International Conference on Artificial Intelligence in Education. pp. 14–19. Springer (2018)
2. Benedetto, L., Cremonesi, P., Parenti, M.: A virtual teaching assistant for personalized learning. arXiv preprint arXiv:1902.09289 (2019)
3. du Boulay, B.: Artificial intelligence as an effective classroom assistant. *IEEE Intelligent Systems* **31**(6), 76–81 (2016)
4. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *Journal of machine learning research* **12**(Aug), 2493–2537 (2011)
5. Eicher, B., Polepeddi, L., Goel, A.: Jill watson doesnt care if youre pregnant: grounding ai ethics in empirical studies. In: AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society, New Orleans, LA. vol. 7 (2017)
6. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., et al.: Building watson: An overview of the deepqa project. *AI magazine* **31**(3), 59–79 (2010)
7. Goel, A.K., Polepeddi, L.: Jill watson: A virtual teaching assistant for online education. Tech. rep., Georgia Institute of Technology (2016)
8. Hill, J., Ford, W.R., Farreras, I.G.: Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in Human Behavior* **49**, 245–250 (2015)
9. Kerlyl, A., Hall, P., Bull, S.: Bringing chatbots into education: Towards natural language negotiation of open learner models. In: Applications and Innovations in Intelligent Systems XIV, pp. 179–192. Springer (2007)
10. Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K.: A survey of design techniques for conversational agents. In: International Conference on Information, Communication and Computing Technology. pp. 336–350. Springer (2017)
11. Shawar, B.A., Atwell, E.: Chatbots: are they really useful? In: Ldv forum. vol. 22, pp. 29–49 (2007)
12. So, S.: Mobile instant messaging support for teaching and learning in higher education. *The Internet and Higher Education* **31**, 32–42 (2016)
13. Sun, Z., Lin, C.H., Wu, M., Zhou, J., Luo, L.: A tale of two communication tools: Discussion-forum and mobile instant-messaging apps in collaborative learning. *British Journal of Educational Technology* **49**(2), 248–261 (2018)
14. Timmis, S.: Constant companions: Instant messaging conversations as sustainable supportive study structures amongst undergraduate peers. *Computers & Education* **59**(1), 3–18 (2012)
15. Ventura, M., Chang, M., Foltz, P., Mukhi, N., Yarbrow, J., Salverda, A.P., Behrens, J., Ahn, J.w., Ma, T., Dhamecha, T.I., et al.: Preliminary evaluations of a dialogue-based digital tutor. In: International Conference on Artificial Intelligence in Education. pp. 480–483. Springer (2018)