Using Relational Inference Engine to Answer Questions

Elias Oliveira[1,2], Jaimel Lima[1,2], Gabriel Dias[1,3] and Juliana Pirovani[1,3]

[1] *Laboratório de Computação de Alto Desempenho*
[2] *Programa de Pós-Graduação em Informática, UFES,*
Av Fernando Ferrari, 514, Goiabeiras Vitória, ES 29075-910 – Brasil
[3] *Departamento de Computação, Universidade Federal do Espírito Santo, UFES,*
Alto Universitário, s/n – Guararema Alegre, ES 29500-000 – Brasil

Abstract

In this paper, we describe a fast chatbot learning methodology so that it can dynamically and automatically learn new information to deal more appropriately with new discourse domains. By our approach, the robot, named *r*AVA, creates internal structures of knowledge while reading more documents, which allow the chatbot to logical reasoning about the world's facts. We experimented with our approach by reading additional documents, extracting family relationship information from the named entity recognized from the free-text contents. After reading these documents, *r*AVA can respond correctly to many questions presented by the users, providing them, by this way, with quick and personalized feedback evaluations.

Keywords

Chatbot, Named-Entity Recognition, Logical Reasoning, Artificial Intelligence

1. Introduction

One of the efforts to keep a virtual assistant pace with the flowing of additional new knowledge is both manually updating the internal knowledge's structures and refactoring the source code containing that knowledge [1]. In this paper, we propose an approach to, mostly automatic, reduce the effort of absorbing knowledge from new domains. Part of the proposed process consists of reading the given fixed content materials and comprehending its content. Another part is the learning process of new pieces of knowledge by Logic Programming with Prolog [2].

A strategy to evaluate whether someone effectively comprehends a subject is by asking them questions about a particular content already read [3]. Nevertheless, with the questions formulated by free-text, we face the style of written questions, language incorrectness, and words variation within the formation of the sentences and the words acquired from the reading content. These are problems a virtual assistant must also face to emulate a better human interaction, and thus, to answer users' questions.

According to the taxonomy given by Grudin of conversational bots [1], it is possible to list three different models: (1) *virtual companion* engages on any topic keeping a conversation going; (2) *intelligent assistant* takes on any topic as well but aims at keeping conversations short;

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY
elias@lcad.inf.ufes.br (E. Oliveira)
http://www.inf.ufes.br/~elias/ (E. Oliveira)
0000-0003-2066-7980 (E. Oliveira); 0000-0002-3727-4158 (J. Pirovani)

(3) *task-focused chatbots* have a narrower range and go deeper, yet brief conversations are their goal. *r*AVA is a robot, or a chatbot, which suits mainly in the third category. It can work as a teaching assistant, and act as a companion of the student along its learning process path [4]. Its conversational skills are not the aim to discuss in this report.

This robot can read free-form text material, formulate questions out of the reading material, and interact with the student to answer and ask friendly questions on the subject. These skills are to promptly and personally evaluate students' knowledge. *r*AVA provides thus the instructors with a detailed diagnostic of the students' learning progress on that subject [5].

To achieve our goal of providing the *r*AVA with new additional knowledge, an algorithm based on that in [6] was used to formulate questions on the reading documents and applied in previous work in [4]. In the current paper, we go beyond that approach discussed in [4]. Firstly, we still read the documents of the interested domain and extract Name-Entities (NE)s [7] we can find within them to build the robot's knowledge-base. In [6], we were able to extract a neat knowledge-base with more than 6,917 questions and answers from a Brazilian-History school book. Secondly, for the sake of our discussion later in this paper, we also recognize only the fatherhood and motherhood relationship between these NEs. With these elements at hand, we automatically create our Prolog knowledge-base of facts, augmenting this way the knowledge of *r*AVA about family-related relationships. Note, we discuss only family relatives' relations, but this approach can also be applied to many other kinds of relations.

Therefore, our contributions in this paper are (a) to discuss a proposed strategy to provide *r*AVA with a quick formed great amount of knowledge better structured rather than the usual flat *frequently asked questions* (FAQs) used in most chatbots, usually structured into the Artificial Intelligence Markup Language (AIML) [8, 9], or similar format. AIML is a language created by Dr. Richard S. Wallace during 1995–2000, and it is similar to the *eXtensible Markup Language* (XML)[1]. Into AIML is placed the possible question the user may inquire, and the respective answer to it as a template. It is also possible to use wildcard characters to augment the flexibility of this language. For this task, our approach is based on the ability we built in for *r*AVA to read a textual document and generate a set of questions out of free-form text such as that discussed by the authors [6]. By being an automatic construction body of knowledge, this reading ability gives *r*AVA great flexibility to easily and quickly be at different domains. In short, we greatly reduce the amount of training data to usually set a bot up ready, and running – it suffices to read a text document to increase the bot's knowledge; (b) Finally, we also show that *r*AVA can reason over relational facts existent over its knowledge base. Both the AIML factual templates base and the relational portion of knowledge coexist within our *robot brain*.

In this paper, our focus is on assessing the ability of *r*AVA to understand a text passage and answer a series of questions that appear in a conversation and related to what was previously read and turned into knowledge to the chatbot. Thus, we want to know how good *r*AVA is to be asked and answer questions related to a given instructional material. For this, we will measure how many times *r*AVA was not able to correctly answer questions made by the user about that given material, measuring by this way the level of knowledge issued by *r*AVA.

The remainder of work is organized as follows. Section 2 presents a brief revision of some related works found in the literature. Section 3 presents the methodology used to identify NEs,

[1]https://en.wikipedia.org/wiki/XML

the Relations (R)s between them, and how we transform the triples $\langle NE_i, R, NE_j \rangle$ into a Question (Q) sentence. The results are presented in Section 4. Section 5 presents the conclusions and future work.

2. The Challenge of Teaching a Bot

The use of a chatbot as a student's companion is described by [10]. the purpose is to have a machine to promptly converse in English with students to practice and improve their level in that language. Unlike the ELIZA approach [11], The author in [10] explores all the words in the user sentence to infer meaning. The user sentence is transformed into a particular XML structure – the author calls it NLML, the user facts. Their implementation improved in many functionalities. For instance, (1) The *common sense knowledge* relating *Student* to a type of *person*, *Stupid* is one antonym of *clever*, and so forth, in user's sentences; (2) The *textual entailment* mechanism to transform sentences into another with similar meanings. That is to augment the matching rating and better deal with the user sentences; (3) The *natural language database* to store common sense atomic facts, in the NLML format, about general knowledge, which can additionally be extended through users' talking.

All these strategies aim to give the conversation more fluency, and the reasoning is done merely on grammatical language structure, these systems do not infer, but merely answer the facts. They use symposium, hypernym, hyponym, and sometimes by using the Wordnet database. The reasoning and inference according to the logical rules are not tackled in its work. The author says that *the laborious transformation from natural language into exact logical language and vice-versa seems to be only done by the logician experts* [10].

One can find considerable advancements already achieved within the area of *Question Answering Systems* (Q&A). In this area of research, the focuses are solely to answer, correctly, the incoming question. There is no interest in conducting a fluent conversion during the user interaction. The paper proposed by [12] is closely related to what we discuss in the current work. Although the authors suggest the results of their work as having *higher potential to build applications such as FAQ chatbots*, and other intelligent systems, they did not go beyond their Q&A application. Their work proposed a combination of Deep Learning Neural Network (DNN), Word2Vec (W2V), and Prolog to build the learning models. Prolog is used for symbolic processing. Although the authors pointed out the ability to learn with a small amount of training data as one of the advantages of using the Prolog inference engine, the DNN and W2V needed a considerable amount of training data to learning Prolog rules. Furthermore, they say that *most of the conventional question answering systems are not entirely capable of inferring from a large amount of information on the Web containing unknown data*, which is resolved by the Prolog rules.

The authors [13] presented and discussed a chatbot based on the platform [14] to support a software testing university class. The authors do not explicitly say how the knowledge database was created. Therefore, we suppose it was manually crafted into the AIML language used by the *PandoraBots* platform.

Rather than the previous applications, different use of a chatbot is the work discussed in [15]. The idea is to translate trigonometric expressions questions written in natural language, entered by students, into equation models, which will be processed mainly *in the context of semantic interpretation and language mapping natural for formal representations of mathematical problems*. The effort focuses on natural language interpretation. The errors during parsing are the approach's drawback. They manually formulated 79 Pythagorean theorem questions, 40 inverse function questions, 123 basic function questions and 17 triangle similarity questions to training and test a *Long Short-Term Memory* neural network (LSTM) responsible for the questions parsing – (*reasoning module*) [15, Fig. 2]. Note that the knowledge of the chatbot is the mathematical solvers they can aggregate to the *resolution module* block [15, Fig. 2]. Hence, this chatbot reasons not on textual facts, but mathematical domains.

A chatbot named ChatterEDU was also developed by [16] to answer or ask questions in Portuguese about a basic education text on the Geography subject. From an input text composed of simple sentences, the questions and answers are generated in the chatbot AIML knowledge base, as proposed in [4]. However, the questions and answers are generated by observing the semantic role. A morphosyntactic analysis is made and the semantic roles of each word are obtained. Seven semantic roles are treated (agent, place, temporal location, temporal origin, extent or quantity, period, and theme), from which is defined what pronoun or adverb should be used to elaborate the question. Although the limitations as to the type of sentence and semantic roles treated, it is possible to generate questions and answers from any text composed of simple and grammatically correct sentences.

3. Reading the Bot's Knowledge

In recent work [4], we proposed an architecture to deal with the learning process for a chatbot. The proposed approach provides a robot with means of acquiring its knowledge by reading free-text documents and extracting elements such as *NE*s and *R*s between entities. That is the novelty of the proposed architecture. The new knowledge is acquired straight from reading texts, not only necessary from prior training datasets. The proposed architecture is presented in Figure 1 and is also the basis for the following experiments in this paper.

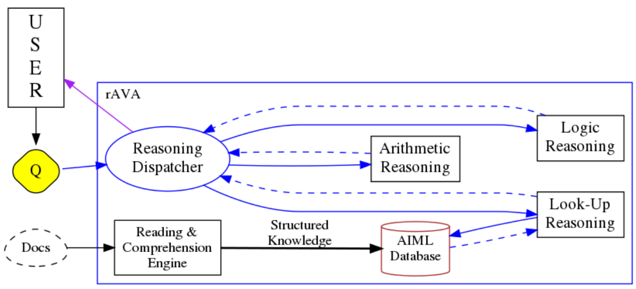


Figure 1: The *r*AVA Architecture

The user submits a query Q to the bot via the *Reasoning Dispatcher*, in short, Dispatcher, representing the front-end layer presentation of our bot. For the experiments discussed in this paper, we will turn available a webpage. But this layer can be adapted to work with different kinds of devices like tablets, smartphones, PCs, Moodle, Telegram, Facebook, and so forth.

So far, we have designed three forms of reasoning for our chatbot: a) the arithmetic processor to allow our chatbot reasoning over numeric values; b) the conventional flat factual QA, which will be treated by the AIML reasoning processor, and c) the logic processor to allow *r*AVA infers over logical or relational facts of its knowledge database.

Arithmetic Reasoning can range from a simple calculator to a sophisticated mathematical solver like that proposed in [15]. The Dispatcher is responsible for classifying the queries and sending them to a more appropriate reasoner. Therefore, Dispatcher preprocesses the original query, turns it into another form to fit into an AIML response entry, or sends it to another reasoner module otherwise. Those queries classified and sent to the Look-Up Reasoning are because their respective answers might be present in the AIML knowledge database.

The ongoing feature we are exploiting in our research is the use of the *Reading Comprehension Engine*. Inside this module, we already extract NEs [17] to build a Q&A, as discussed in [4]. In this paper, we are interested in a particular relational structure also extracted straight from a document. For our experiments, we used the Bible's Mathews book, Chapter 1. We discussed our experiments in Section 4. The chosen document was because of the abundance of expressed facts regarding family relations: fatherhood and motherhood.

3.1. From Original query Q to the transformed Q'

In [4], we show the *r*AVA's ability to read a history book, extract NEs [18], and generate questions and their respective answers to populate its knowledge database. Now we discuss an improvement to deal with more elaborated questions. That is the query expansion for its classification and selection of the suitable reasoning module to be sent. Although our experiments tackle only a particular relational query structure, this strategy can easily be generalized to embrace other similar relations.

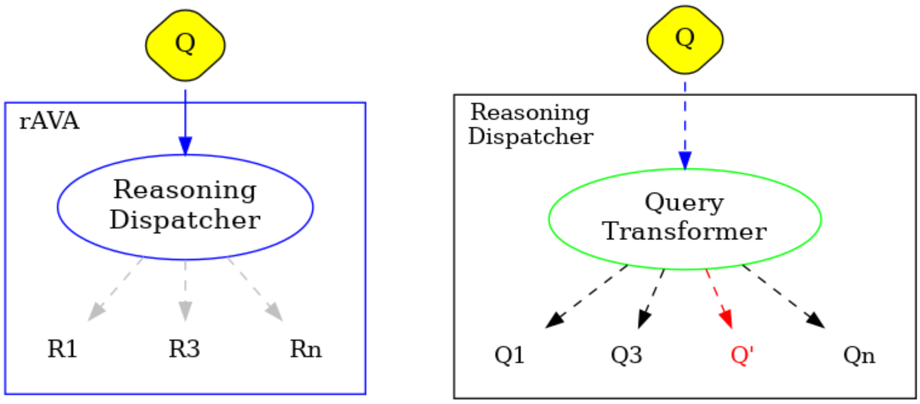
When one posts search questions to the *r*AVA these questions must first be annotated, untangle into small parts, restructured in a way to be sent to the right knowledge database to answer them. Let us consider a query Q, for instance: *Who is the mother of Jesus?*[2]

Firstly, we need to recognize, if existent, the NEs into the sentence, annotated them in order to restructure the sentence into another format, Q'. When processing the previous query sentence by the *spaCy* tool, we can transform that original sentence into the following form:

<N>Who</N> <REL>is the mother of</REL> <PERS>Jesus</PERS>?

Thus, after the annotation, we can deal with the entities to reformulate the original question into another structure suitable to be sent to the varieties reasoning modules, as we can see in Figure 2a. The query is first transformed into various formats, as depicted in Figure 2b, to try each of the available reasoning modules. Figure 2 reproduces the processing section between the query Q and the module responsible for processing it and finding out to whatever type of reasoning $\{R_1, R_2, \ldots, R_n\}$ the already transformed query should be sent, in Figure 2a. R_1 could be, for instance, the *Arithmetic Resoning* module, R_3 the *Look-Up Reasoning*, and so forth.

[2]The challenge is to learning the *Jesus Christ*'s genealogy by reading directly from free-text: ⟨ https://en.wikipedia.org/wiki/Genealogy_of_Jesus⟩



(a) The original query Q (b) New queries transformed from Q

Figure 2: The query transformations' flow: (a) the user query and (b) the automatic query expansions

Whereas in Figure 2a, we have the flow thru the Reasoning Dispatcher up to the Reasoning Modules built in the *r*AVA knowledge bank of inference, in Figure 2b, we explode the first step inside the Reasoning Dispatcher. We may apply various techniques to reformulate a query into another form. All this is to try to capture the user intent. However, in this paper, we discuss only one type of reformulation. Our focus in this paper is on when that reformulation Q' results in a logical call to the *r*AVA's *Logic Reasoning* module.

The next step is to transform the annotated sentence into a predicate structure used by Prolog. Thus, query Q is transformed into the predicate Q' form using the morphosyntactic classifications and the dependency tree generated by the spaCy. We obtained the predicate name combining the auxiliary verb *is* and the substantive *mother*. The first parameter for the predicate is *WP*, a classification also generated by spaCy for pronouns in questions related to the person, in this case, *Who*. Prolog considers arguments with the first letter in uppercase as variables. We obtained the second parameter by the second NE found in the sentence *jesus*. Therefore, Q is turned into the structure Q' as such:

is_mother(WP, jesus).

In our case study, we need a structure to invoke the logical engine to tell whether the *Logic-Prolog Clause* is satisfied and then restructure back a syntax-valid answer to the user. Thereby, the *<N>Who</N>* is transformed into a logical variable, *WP*, not yet instantiated. Jesus, *<PERS>Jesus</PERS>*, is recognized as a person name and thus is placed as an atom for a Prolog clause. The *Prolog Horn-Clauses* [2] are extracted from the relations recognized between any pair of entities found in any sentence. It is the most tricky part of our approach. Firstly, during relations recognition and correct extraction. Secondly, when transforming these relations into a new structure to match to a Prolog Horn-Clause, whether it is a valid one into the *r*AVA's knowledge database. The following Prolog goal, is a possible transformation of the previous given query. The relation was transformated into a matched predicate in the experimental dataset used for testing our approach in this paper.

In sum, in Table 1, the original query is transformed into Q', among some other transformations

Q	*Who is the mother of Jesus?*
Q'	**is_mother**(WP, jesus).

Table 1
The query Q transformed into Q'

which did not succeed in matching to a Prolog predicate into the knowledge database.

4. The Experiments and Results

Besides those previous experiments presented in [4], we improved our bot with the ability to extract relational knowledge, again straight from text documents. Therefore, to evaluate this improvement, we created a set of questions to enter into the *r*AVA interface and test their results. These questions are listed in Table 2. We planned these queries with the only objective to illustrate how our architecture functions. It is out of scope of this work to discuss any other ordinary question with syntax problems, semantically identical but unmatched expressions, or other issues. To these problems, our robot will answer, for instance, a default answer: *I am sorry, but I do not have enough information to answer this query!*

Figure 3: The *r*AVA's web-interface for experimentations

An experimental web-interface was used for these evaluations, as shown in Figure 3 [3]. The users place their query in the blank-box and press the *Ask* button (see in Figure 4) to send this requisition to the *r*AVA's internal system (see Figure 1, in Section 3).

Figure 4: The *r*AVA's web-interface for experimentations

[3] http://www.eliasdeoliveira.com.br/rava/index-form.html

In Figure 5, we show the answer returned by the *r*AVA's logical-reasoning over the Bible chapter recently acquired knowledge. In this case, the positive answer $A+$ is returned (see Table 2).

Figure 5: The *r*AVA's web-interface for experimentations

We listed, in Table 2, some of the queries Qs we used to test our approach with their respectively positive $A+$ and negative $A-$ answers. For instance, given the fourth sentence: *Who is Jesus' grandfather?*, the answer for this query is *The grandfather of Jesus is Jacob!* is a positive answer for the query $Q4$, whereas a negative reply could be: *I am so sorry, but I have no information about who the grandfather of Jesus is*. The first three sentences, in light-green color, are queries resolved by the basic inference Prolog engine. Nevertherless, we need a more sophisticated inference model to solve the two lastest queries.

Q	Sentence	$A+$	$A-$
1	*Who is the* **mother** *of Jesus?*	*The* **mother** *of Jesus is WP*	*I am so sorry, but I have no information about who the* **mother** *of Jesus is*
2	*Who is* **father** *of Jesus?*	*The* **father** *of Jesus is WP*	*I am so sorry, but I have no information about who the* **father** *of Jesus is*
3	*Who is* **father** *of Jose?*	*The* **father** *of Jose is WP*	*I am so sorry, but I have no information about who the* **father** *of Jose is*
4	*Who is Jesus' grandfather?*	*The* **grandfather** *of Jesus is WP*	*I am so sorry, but I have no information about who the* **grandfather** *of Jesus is*
5	*Is Mary married?*	*Yes, Mary is married!*	*I am so sorry, but I have no information about to conclude this.*
		Yes, Mary is married to WP!	*... idem ...*

Table 2
Examples of Q and possible syntax-valid answers $A+$, or $A-$

Although we showed here a pathway to solve a difficult problem of logical-reasoning having a free-text as the origin of the query, this is only an ongoing work. There is a lot more further work to get a robot at a reasonable level to mimic a decent human-machine conversation. In our approach, we still need to work further on the construction of the return back the answers for given queries, as exemplified in Table 2. The flux from query Q to the *Prolog Predicate* is an important candidate for further improvement. The transformation of a *recognized relation* into a viable Prolog predicate is a sensible task. That is dependent, for instance, on the application, user contextual community, and so on. In our experiments, *is_father* could also be transformed into *is_father_of* or just *father*. This is an open problem in our approach that we will tackle in the future work.

Another future important work subject is to deal with disambiguation. In Table 2, in the line before the latest one, Jesus is also named by *jesus_christ*. That is a simple task for human beings to resolve, but still hard for a machine to reconcile this disambiguation issue. For instance, in the document used for these experimentations, the author used *poetic license* to say that David is the father of Jesus. This fact would conflict with another that says Joseph is the father of Jesus. To solve this particular problem a simple solution could be that where one inserts a new contextual *rule* to tell that a grandfather of any level could also be named the father of someone. Note that, in doing that, we are *logically inferring synonyms* for terms in the textual documents.

5. Conclusions and Future Work

Logical-Reasoning is a valuable tool to solve a vast range of artificial intelligence problems. In this paper, we exploit this inference tool to reasoning over family genealogy extracted from free-text documents. In this paper, we proposed a strategy to annotate and recognize NERs within this query, transform it into a valid Prolog query. This new logical-query is processed by the Prolog engine and its result is transformed back as an answer to the user. Note that the identification of NERs within the text of the documents [18] is the core part of our approach. Based on these entities, we restructure question-sentences for the logical-knowledge database.

We built this process into our robot reasoning architecture. Besides, our proposal generates a knowledge-base by reading free-text documents, reducing the manual effort of implanting into robot its new knowledge. Part of this process still needs improvements, as discussed in [6]. But the possibility of populating a knowledge-base using content learning from a subject already expressed in documents and books is promising in building a cross-domains intelligent robot.

This approach was first used to test students' knowledge from a history book, now *r*AVA is able to deal with textual relational structures and also formulate questions from them. Therefore new type of questions can now be applied to evaluate the students. Besides, this approach of using a robot to evaluate students is cheaper, scalable, and can be individualized, which benefits the students' learning process.

We tested our approach by reading a free-text document, generating a knowledge database from the extracted facts, and allowing our robot *r*AVA to reason both on a history book facts [4] and over a biblical family genealogy. These experiments opened up new fronts for future research, such as deal with disambiguation, unmentioned genre, and much more. Another plan is to apply this methodology to tackle this problem in another documents' domain.

References

[1] E. Adamopoulou, L. Moussiades, An Overview of Chatbot Technology, in: I. Maglogiannis, L. Iliadis, E. Pimenidis (Eds.), Artificial Intelligence Applications and Innovations, Springer International Publishing, Cham, 2020, pp. 373–383.

[2] W. F. Clocksin, C. S. Mellish, Programming in PROLOG, 5 ed., Springer Science & Business Media, Germany, 2003.

[3] S. Reddy, D. Chen, C. D. Manning, CoQA: A Conversational Question Answering Challenge, Transactions of the Association for Computational Linguistics 7 (2019) 249–266.

[4] E. Oliveira, M. Spalenza, J. Pirovani, *r*AVA: A Robot for Virtual Support of Learning, in: 20$^{\underline{th}}$ International Conference on Intelligent Systems Design and Applications – (ISDA), Springer, Springer International Publishing, On the WWW, 2020, pp. 102–113.

[5] E. Hunt, Mining for Patterns of Incorrect Response in Diagnostic Assessment Data, in: Handbook of Educational Data Mining, CRC Press, 2010, pp. 427–439.

[6] J. Pirovani, M. Spalenza, E. Oliveira, Geração Automática de Questões a Partir do Reconhecimento de Entidades Nomeadas em Textos Didáticos, in: XXVIII Simpósio Brasileiro de Informática na Educação (SBIE), SBC, Ceará, CE, 2017, pp. 1147–1156.

[7] H. Zhang, F. Boons, R. Batista-Navarro, Whose Story is It Anyway? Automatic Extraction of Accounts from News Articles, Information Processing & Management 56 (2019) 1837 – 1848.

[8] G. De Gasperis, I. Chiari, N. Florio, AIML Knowledge Base Construction from Text Corpora, in: Artificial intelligence, evolutionary computing and metaheuristics, Springer, 2013, pp. 287–318.

[9] R. Wallace, The Elements of AIML Style, Alice AI Foundation 139 (2003).

[10] J. Jia, CSIEC: A Computer Assisted English Learning Chatbot Based on Textual Knowledge and Reasoning, Knowledge-Based Systems 22 (2009) 249–255.

[11] J. Weizenbaum, ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine, Communications of the ACM 9 (1966) 36–45.

[12] H. Honda, M. Hagiwara, Question Answering Systems With Deep Learning-Based Symbolic Processing, IEEE Access 7 (2019) 152368–152378.

[13] L. H. C. Rosa, E. Lemos, G. Bernardi, R. Medina, Ensino de Testes de Software por Meio de Digital Storytelling e Chatterbots, in: XXVIII Simpósio Brasileiro de Informática na Educação (SBIE), volume 28, 2017, p. 797.

[14] PandoraBots, http://www.pandorabots.com, 2021. [Acc. at 31-Aug-2021].

[15] N. L. Kuyven, V. J. Barros Vanzin, C. A. Antunes, A. Cemin, J. L. T. Silva, L. M. R. Tarouco, Assistente Conversacional para Resolução de Problemas Trigonométricos em Linguagem Natural, Revista Brasileira de Informática na Educação 28 (2020) 208.

[16] J. Martins, C. Martins, Geração de Perguntas e Respostas para a Base de Conhecimento de um Chatterbot Educacional, in: Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology, 2017, pp. 22–26.

[17] E. Oliveira, G. Dias, J. Lima, J. C. Pirovani, Using Named Entities for Recognizing Family Relationships, in: 8$^{\underline{th}}$ Symposium on Knowledge Discovery, Mining and Learning – *KDMILE*, SBC, Rio de Janeiro, RJ, 2021.

[18] J. Pirovani, E. Oliveira, Studying the Adaptation of Portuguese NER for Different Textual Genres, The Journal of Supercomputing (2021) 1–17.