

# Chatbot Optimization using Sentiment Analysis and Timeline Navigation

Otimização de Chatbots utilizando Análise de Sentimento e Navegação em Linhas Temporais

Wagner da Silva Maciel Sodré<sup>1\*</sup>, Julio Cesar Duarte<sup>1</sup>

**Abstract:** A chatbot or conversational agent is a software that can interact or “chat” with a human user using a natural language, like English, for instance. Since the first chatbot developed, many have been created but most of their problems still persist, like providing the right answer to the user and user acceptance itself. Considering such facts, in this work, we present a chatbot-building framework that considers the use of sentiment analysis and tree timelines to provide a better chatbot answer. For instance, as presented in our experiments, the user can be addressed to a human attendant when its sentiment is very negative, or even try another branch of the tree timeline, as an alternative answer, whenever the user sentiment is less negative.

**Keywords:** Chatbot — Framework — Sentiment Analysis — Timeline Tree

**Resumo:** Um chatbot ou sistema conversacional é um software que pode interagir ou “conversar” com um usuário humano usando linguagem natural, como o Inglês, por exemplo. Desde o primeiro chatbot desenvolvido, muitos foram criados mas a maioria dos seus problemas ainda persistem, como prover a resposta correta para o usuário e o próprio aceitação do usuário em usar esse tipo de tecnologia. Considerando esses fatores, nesse trabalho, é apresentado um framework para a criação de chatbots que considera a utilização de análise de sentimentos, bem com de árvores de linha temporal para prover uma melhor resposta pelo chatbot. Por exemplo, conforme apresentado nos experimentos, o usuário pode ser encaminhado para um atendente humano quando o sentimento for muito negativo, ou até mesmo tentar outro ramo da árvore de linha temporal, como resposta alternativa, se o sentimento do usuário for menos negativo.

**Palavras-Chave:** Chatbot — Framework — Análise de Sentimento — Árvore de Linha Temporal

<sup>1</sup> Post Graduation Program in Systems and Computing, Military Institute of Engineering (IME), Rio de Janeiro - Rio de Janeiro, Brazil

\*Corresponding author: wsodre@ime.eb.br

DOI: <http://dx.doi.org/10.22456/2175-2745.125825> • Received: 11/07/2022 • Accepted: 03/01/2023

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

## 1. Introduction

Since the beginning of computing, there has always been a wish of simulating the behavior of another human being. From this wish, ELIZA has been created, the first chatbot in history that simulated the behavior of a psychologist, changing the structure of a sentence sent by the user, and returning it as a question [1, 2]. A chatbot or conversational agent is a software that can interact or “chat” with a human user using a natural language, like English, for instance [3].

After ELIZA, there was a lot of investment in the development of new, smarter chatbots, although without success, mainly due to the technical limitations at that time. However, in the last years, with the advance of Computing and Artificial Intelligence, this scenario has completely changed [4, 5] allowing the emergence of chatbots for many areas and the optimization of tasks such as customer service.

This fact happens due to the emergence of technologies for Processing and Understanding Natural Language, Sentiment Analysis, and Deep Neural Networks, among others, that are able to better identify the real objective of a user, in a way that a computer can be trained to answer those questions in a smarter way.

Chatbots still do not understand dialogues as well as a human being, but they can make simple tasks and, when they do not know the answer, they can redirect it to a human to continue the dialogue. Due to this big chatbot error risk, most of them usually send to the users a fixed list of options, where they can navigate, like an automatic telephone answering system, but using text instead of audio [6, 7].

This leads to chatbot acceptance being very low nowadays, mainly because often users cannot solve their problems directly through the chatbot without the help of a human, which

causes users' frustration and, consequently, the unwillingness to use this type of service, since it is seen as ineffective. Research shows that around 75% of customers experience poor customer service and that generating meaningful, lengthy and informative responses remains a challenge [3, 6].

The performance of this type of technology is one of the main facts for its low acceptance since chatbots tend to get lost in the flow of conversation, thus not being able to conduct a real conversation with a human. Even though language interpretation is processed by statistical models, most dialog systems released still use manual resources or rules for representing the states and actions' space, detecting intents, and filling in spaces. This not only makes the system more expensive and slower to respond in a real conversation, but also limits its use in other domains [8, 9].

Another important fact is the lack of empathy that chatbots have. A chatbot should be able to interact with a human naturally or it should present itself as a robot, displaying the possible options to the user. When a chatbot does not meet these conditions, user acceptance also drops dramatically [3].

It is important to reinforce that if a chatbot is going to act like a human, it must be clear to the user that he is talking to a machine and not a human. As much as artificial delays or indicators like "typing..." can be used to make the dialogue more familiar to the user, impersonating a human risks increasing the distance between the service and the user [10].

Sending a message to a machine that may not understand what a user is saying can be a frustrating experience, especially when a machine hides its flaws in an artificial dialog to keep the conversation "natural" or "human-like". This hides points of failure in the conversation and makes the user feel less in control of the interaction. This does not imply that chatbots should not have a personality or take advantage of humorous and emotional responses to produce more attractive interactions for the user [10].

As suggested by [Nuruzzaman e Hussain](#)[6], chatbots often lose context, and this could be optimized by using trees to create adaptive dialog timelines. Furthermore, [Lee et al.](#)[11] suggests that sentiment analysis improves communication between the user and the machine.

Therefore, the objective of this work is to propose a framework for building chatbots that use timelines and sentiment analysis. Such chatbots should not get lost in context, nor be repetitive, in addition to use sentiment analysis to detect whether the user experience is good. Otherwise, they must redirect the user to a real attendant, as a last option.

The rest of this work is organized as follows. Section 2 presents the main theoretical aspects used in by our proposed methodology. Section 3 presents the related work. Section 4 presents the methods that were used. Section 5 describes the results of the framework and a discussion about it, and finally, section 6 concludes the research and suggests future works.

## 2. Background

This section provides the main concepts that are related to this work, like the concepts of Framework, Library, Chatbot, Natural Language Processing (NLP) and Sentiment Analysis.

### 2.1 Framework

As stated by [Riehle](#)[12], a framework is a model of a particular domain or an important aspect of it. A framework can model any domain, be it a technical domain like distribution or garbage collection, or an application domain like banking or insurances. It provides reusable design and implementations for customers that contain common functionalities across a variety of applications and must belong to the same domain.

There are four main points in a framework development [12]:

- It must provide a solution to a diversity of similar problems;
- It must use a set of classes and interfaces that shows how to decompose a diversity of problems;
- It must use objects of these classes in a collaborative form to fulfill its responsibility; and
- It must provides a set of classes that must be flexible and extensible, in order to be able to develop a variety of applications with minor effort that specifies only the particularities of each application.

### 2.2 Library

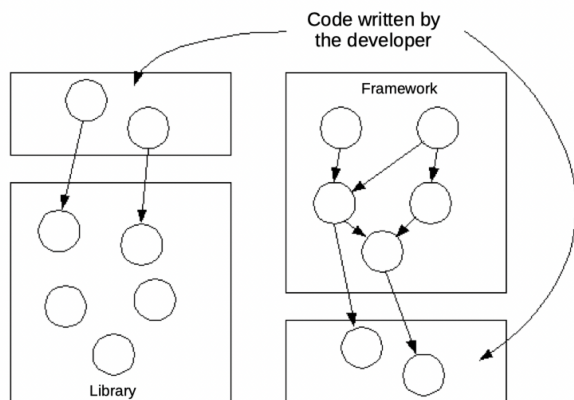
As stated by [Levine](#)[13], in the 1940s and early 1950, programming shops had actual code libraries containing reels of tape (or later, decks of cards) that a programmer would visit and select routines from to load with his program. Once loaders and linkers started to resolve symbolic references, it became possible to automate the process by selecting routines from libraries that resolve otherwise undefined symbols.

A library file is, fundamentally, no more than a collection of object files, usually with some added directory information to make it faster to search. In a Library, each class is unique and independent of others classes, differently from a Framework, where dependencies between classes are built-in, as illustrated by [Figure 1](#)

### 2.3 Natural Language Processing

Natural Language Processing (NLP) refers to the process used to collect external data to a corpus or collect new data in a dialogue between a system and an user. This involves data preparation and transformation of common texts in natural language into an input that machines can understand [15, 16].

The data of this process can be used in two components. First, it is used as a knowledge base of the system or the database, in order to build a system, like a chatbot. Second, as data that the system collects during a dialogue with the user and that is used as information that helps the machine in

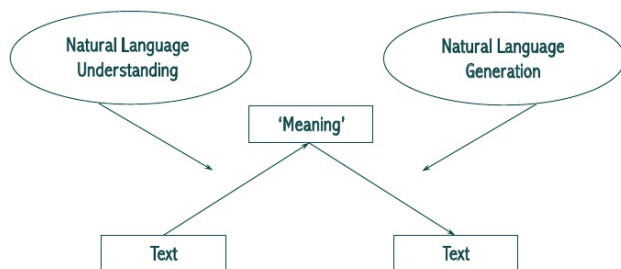


**Figure 1.** Difference between a library and a framework. Adapted from SauvÉ[14]

the understanding process, during the processing step. NLP extracts mainly the (semantic) meaning of the text [15, 16].

As illustrated in Figure 2, NLP Process can be divided into 2 subprocesses, Natural Language Understand (NLU), which is the part that translates the natural language to the machine, and Natural Language Generation, which is the machine answer to the user.

$$NLP = NLU + NLG$$



**Figure 2.** The NLP Process. Source: [17]

Natural Language Understanding (NLU) is one of the most important steps in an automated dialogue system and is responsible for mapping the user input in natural language in a way that a machine can understand it, so the next steps can be processed. Many approaches suggest the semantic analysis of the user input, and among them, the most used is a grammar-based analysis [15, 18].

Conversely, Natural Language Generation (NLG) is the NLP part that is responsible for the generation of machine answers for the user, based on the information that each conversation has, like user intentions and system’s knowledge. This answer is used in a form that provides some kind of information to users or asks a question, demanding more information about the users’ intent. To be able to accomplish this, NLG needs a knowledge base and understanding of what the user says [15].

### 2.3.1 Sentiment Analysis

Among the diversity of information available on the internet, one that is very useful is the sentiment, or, more precisely, the opinion that a person expresses about a subject. For example, knowing your own reputation or the reputation of the rivals of a company can be very valuable in the development of a product, marketing, or management of the relation with the client [19].

The essence of sentiment analysis is identifying how sentiment can be expressed in a text, in a **positive** or **negative** way. Although it may seem like a simple task, many compound sentences may contain more than one sentiment. For example, in the affirmation, “The product A is good but expensive”, there are two affirmations. The first is “The product A is good”, which shows a positive sentiment and the second is “The product A is expensive”, which shows a negative sentiment. For this reason, the sentiment of the context may not be obtained, and, instead, we try to extract parts of each affirmation and the sentiment of each part [20].

Through the application of lexicon and rule-based sentiment analysis algorithms, it is possible to extract the positivity of each sentence, classifying it, for instance, in a real value between -1 and 1, where -1 is a completely negative sentence, 0 is a neutral sentence and 1 is a completely positive sentence.

In Table 1, it is possible to observe some examples of sentences and their respective positivities, using the proposed algorithm implemented in Python’s VADER (Valence Aware Dictionary and sEntiment Reasoner) Sentiment Library, one of the most popular sentiment analysis libraries, which is also built-in the NLTK Framework [21].

Sentence	Positivity
I would like to check my account activity	0.3612
lovely	0.5859
you’re not helping me at all	-0.2235

**Table 1.** Example of sentences and their positivities

### 2.3.2 Natural Language Processing Libraries

NLP libraries are tools that process text and obtain its knowledge, for example, the subject to which users’ sentiment refers. After training using texts in the desired language, the tool is able to obtain the topic of the text, classify the words it refers to and its subject, among other features [15, 16].

The five most used libraries in the market according to Omran e Treude[22] and Kosmajac e Kešelj[23] were selected for this work, and a comparative study was made.

This comparative study was carried out measuring some key characteristics for the project, like Portuguese Language and Sentiment Analysis support. Other features were also evaluated, such as the built-in sentiment analysis model and the native sentiment classifier. The result of this study is presented in Table 2.

	NLTK [24]	SpaCy [25]	Polyglot [26]	Google Syntaxnet [27]	Stanford CoreNLP [28]
Portuguese Language Support	✓	✓	✓	✓	
Sentiment Analysis Support	✓	✓	✓		✓
Sentiment Analysis Model built-in	✓		✓		
Native Sentiment Classifier	✓				

**Table 2.** Comparison between the five main NLP Libraries

## 2.4 Chatbots

There are many ways to make human-computer interactions. One of them is through natural language, which also has many sub-approaches and objectives. Among them is chatbots, which have the main goal of making the computer dialogue in a natural way the closest possible to a real human. This dialogue generally is based on tasks, like when the computer is considered an assistant that receives specific tasks, such as internet search, file organization, or scheduling task management [29].

Nowadays, the biggest existing challenge to chatbots is to maintain the context and understand what users say in order to better answer them. Most chatbots that exist still work only with the identification of a simple pattern in the user input and then, try to find a built-in answer that combines with that input. This approach, however, does not result in a completely satisfying conversation or direct the conversation towards any specific purpose [29].

Chatbots identify entities and intentions of the user text, so that it tries to understand the meaning of the text. The intentions are the user objectives when chatting, and the entity gives the meaning and makes the goals of the user more certain. For example, when one informs a chatbot the sentence “cancel the telephone service”, the intention is the cancellation and the entity is the telephone service. Likewise, in the case “I want to buy clothes”, the *intention* is to **buy** and the *entity* is **clothes**. The entity types and intentions are configured by the chatbot developer, based on its purpose [30].

The main problem of using a chatbot framework, like Google’s Dialogflow, is that its code is limitedly customizable and features like timelines and sentiment analysis cannot be directly implemented.

## 2.5 Intention Detection Techniques

To detect user intent, the TF-IDF measure can be used, which defines the relevance of a word or phrase within a collection of texts. This can be very useful to get the intent of a sentence and, with it, be able to define the right answer.

The Term Frequency (TF) is the division of the number

of times that a word appears in a text by the total number of words in the text (Equation 1) [31].

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

The Term Frequency,  $tf(t,d)$ , can represent the relative frequency of a term  $t$  within a document  $d$ , where  $f_{t,d}$  is the raw count of a term in a document, i.e., the number of times that the term  $t$  occurs in the document  $d$ . Note that the denominator is just the total number of terms in document  $d$ , counting each occurrence of the same term separately.

The Inverse Data Frequency (IDF) is a measure of how much information the word provides, i.e., if it is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the term  $t$ , which is obtained by dividing the total number of documents  $N$  by the number of documents containing the term  $t$ , and then taking the logarithm of that quotient, as presented in Equation 2 [31].

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

Finally, in Equation 3, we obtain the TF-IDF measure which is simply the multiplication of TF by IDF [31].

$$tfidf(t,d,D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \times \log \frac{N}{|\{d \in D : t \in d\}|} \tag{3}$$

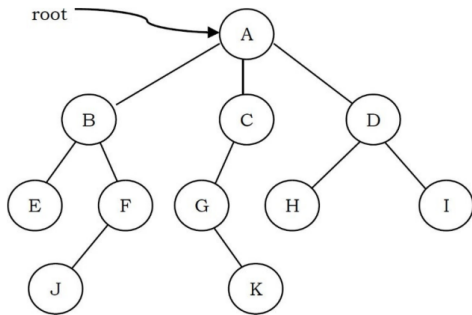
A high value of TF-IDF is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents, where these weights tend to filter out common terms. Since the ratio inside the IDF’s log function is always greater than or equal to 1, the value of IDF (and TF-IDF) is always greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches to 1, bringing the idf and tf-idf closer to 0.

## 2.6 Trees

As defined by Karumanchi[32], a tree is a data structure similar to a linked list but instead of each node pointing simply to the next node in a linear fashion, each node points to a number of nodes representing a nonlinear data structure. A tree structure is a way of representing the hierarchical nature of a structure in a graphical form. In ADT (Abstract Data Type) trees, the order of the elements is not important. If we need ordering information, linear data structures like linked lists, stacks, queues, among others, may be used. An example of a tree is presented in Figure 3.

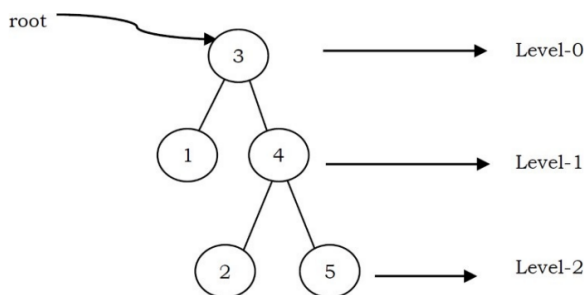
The main components of a tree are:

- The root of a tree is the node with no parents. There can be only one root node in a tree (node A in Figure 3);



**Figure 3.** Basic tree data structure. Source: [32]

- Edges refer to the links from parent to child (all links in Figure 3);
- Nodes with no children are called leaf nodes (E, J, K, H and I in Figure 3);
- Children of the same parent are called siblings (B, C, D are siblings of A, and E, F are the siblings of B in Figure 3);
- A node p is an ancestor of node q if there exists a path from the root to q and p appears on the path. The node q is called a descendant of p (A, C and G are ancestors of K in Figure 3); and
- The set of all nodes at a given depth is called the level of the tree (B, C and D are at the same level in Figure 3). The root node is always at level zero.



**Figure 4.** Tree Level Structure. Source: [32]

One of the most critical aspects of a tree is its level structure, as illustrated in Figure 4. The main tree level concepts in order to better understand it are:

- The depth of a node is the length of the path from the root to the node (depth of G is 2, A – C – G);
- The height of a node is the length of the path from that node to the deepest node. The height of a tree is the length of the path from the root to the deepest node in the tree. A (rooted) tree with only one node (the root) has a height of zero. In Figure 3, the height of B is 2 (B – F – J);

- The height of the tree is the maximum height among all the nodes in the tree and the depth of the tree is the maximum depth among all the nodes in the tree. For a given tree, depth and height return the same value, but for individual nodes we may get different results; and
- The size of a node is the number of descendants it has including itself (the size of the subtree rooted at node C is 3).

A tree flow is defined as the path that is needed to access a node, starting from the root. For example, in Figure 3, to access node F from the root (Node A), it is first necessary to reach node B, and, then, finally node F.

### 3. Related work

This section presents some related works. First, we discuss works that address user acceptance of chatbots, and then, some works that try to solve problems related to this low acceptance.

#### 3.1 User Acceptance

Chatbots user acceptance is very low, since only 45% of users prefer to use chatbots. That is when they are motivated to use the tool, and that number can drop even further when they are not [3].

Users tend to get frustrated and tend not to use chatbots because they are not able to solve user problems, leading to use human attendants [3]. In addition to not being able to solve customer problems, chatbots often miss the context of the conversation, getting lost in the dialogue and further frustrate the user [8].

There are two points that are constantly criticized in studies related to chatbots. First, the machine does not correctly understand what the user wants. The second one is that chatbots tend to be extremely limited to a single domain, and even in this domain, they provide little relevant information to the user, leaving him even more dissatisfied with the service provided by the company [33].

While many recent chatbots in the form of intelligent personal assistants allow free text input through their interface, which invites their users to “ask anything”, there is a surprising empirical limitation on how users actually converse with these agents. This creates a challenge in the development of chatbots, which consists of strong anticipation of what the user must type to the chatbot [33].

For example, upon inferring a decline in user engagement, an agent can immediately employ strategies to reactivate the user. To infer this internal user status, agents rely on the recognition of signals in the users’ behavioral manifestation. For example, staring and attentive feedback (“un-huh”) are signs of engagement. These association rules are an integral part of the computational models underlying adaptive agents [33].

However, most existing work has been based on observations of human-human communications and has aimed to infer

human concepts of interpersonal status such as relationship and trust. Key areas of conversation include feedback, funny jokes, system questions, and usual communicative statements. Through the lens of statistical modeling, rich signals in conversational interactions to infer user satisfaction can be used to develop agents that adapt algorithmic performances and interaction styles [33].

Another important factor overlooked in most chatbots is the tone of the user. Chatbots for customer service often focus only on generating grammatically correct answers, ignoring other factors that can affect the user experience. Many works suggest that the tones used in the responses to users are essential for satisfactory service. For example, a courteous tone has a significant effect on outcome satisfaction and an empathetic tone reduces user stress and results in more engagement [34].

It is observed, for example, that an empathetic tone significantly reduces the negative emotion of users, such as frustration and sadness. In addition, a passionate tone excites users and increases service satisfaction. According to the analysis results, it is possible to identify tones that are beneficial for customer service in order to study the representative words of these beneficial tones. These are interesting and valuable guidelines for future works with possible directions including studies of the effects of agent tones at a finer granularity and how chatbots can affect end-user engagement [34].

### 3.2 State-of-the-art in the problem

In order to analyze the state of the art on chatbots, a systematic literature review related to the research topic was carried out. The search was carried out using the years 2015 to 2020 as a reference. We used IEEEXplore<sup>1</sup> and ACM digital library<sup>2</sup> as main search engines, and we analysed the articles with more citations and relevance for the query “chatbot”. Also, a survey was carried out to verify which NLP tools are the best and how temporality and sentiment analysis influence the dialogue with chatbots.

The analysis of the state of the art sought to identify which works compare technologies used in the development of chatbots and also the opinion of experts.

Ren et al.[35] analyzed more than 700 sources and retrieved 28 primary studies while conducting a systematic mapping study to identify the research questions, characteristics, and metrics used to evaluate the usability of chatbots in experiments. The objective of this work is to determine the state of the art of chatbot usability experimentation. It was concluded that chatbots have several advantages (e.g. they provide real-time response and improve ease of use) and some shortcomings (e.g. the use of NLP, which is ranked as the weakness most in need of improvement). It is important to note that it emphasizes that the increasing interest in this area is very recent, as works did not start to be published until 2018. Also, there was only one chatbot in the research that is equipped with a sentiment analyzer as it discovers items that

best fit users’ needs.

Athota et al.[36] created a chatbot to diagnose some diseases and provide basic details about them. The objective of this work is to create a medical chatbot using Artificial Intelligence that can diagnose a disease and provide basic details about the disease before consulting a doctor. To fulfill this, a chatbot was built that stores the data in a database in order to identify phrase keywords and make query decisions and question answering. Ranking and sentence similarity calculations are performed using n-gram, TF-IDF, and cosine similarity. The score is then obtained for each sentence from the given input sentence and more similar sentences are obtained for a given query. It concludes that the chatbot removes the burden from the answer provider by directly delivering it to users using expert systems while saving their time.

Daniel et al.[37] created a multi-channel and multi-platform chatbot modeling framework called Xatkit, with an objective to provide a set of Domain-Specific Languages that define chatbots (and voicebots or bots, in general) in a platform-independent way. Xatkit also comes with a runtime engine that automatically deploys the chatbot application and manages the conversation logic defined on the platforms of choice. It was concluded that Xatkit is ready to be used in real case scenarios, but still has a lot of room for improvement, such as combining sentiment analysis and behavioral design patterns to create more friendly and effective chatbots.

Jain et al.[30] analyzes that there is a difference between the chatbots’ state of knowledge (context) and the users’ perception of what the chatbots are understanding. The objective of this work is to propose a window in the chatbot interface showing its context for a better understanding of the user. The tool used in this work was the IBM Watson Conversation, which helps in the temporality of the dialogue since the tool has some sort of timeline to guide the conversation. To test this concept, a demonstration software was created based on a pre-existing chatbot from an online shoe store, and sixteen participants were asked to test the interfaces, evaluating points such as effort, task success, and frustration, in grades from one to five. At the end of the research, it was concluded that the performance and user acceptance of the proposed framework was better than the previous one.

Lee et al.[11] assesses that most conventional chatbot models only try to find sentences that are more likely to be related to user input, without taking into account the sentiment of the output sentences. The objective of this work is to propose five different models for scaling or adjusting the feeling in the chatbot response. For this, two metrics were created to assess whether the model’s response was coherent or not. These metrics were combined with two others that are popularly used (Sentiment Classifier Score and Language Model Score). Tests were also carried out with thirty undergraduate students, where they should rate the coherence, grammar and feeling of the answer from zero to five. It was concluded that the best model is the CycleGAN, which can transform a sentence with a negative sentiment into a positive one.

<sup>1</sup><https://ieeexplore.ieee.org/>

<sup>2</sup><https://dl.acm.org/>

Nuruzzaman e Hussain[6] performs a bibliographic review with the most used chatbots today. Its purpose is to list the similarities, differences, and limitations of each one. For this, it compared the eleven most popular applications in terms of features and technical specifications and analyzed more than seventy publications related to the subject in the five years prior to the work. This research has shown that approximately 75% of customers have experienced poor customer service and that generating meaningful, lengthy and informative responses is still a challenging task. It also shows how failures to generate responses affect conversation quality.

In Io e Lee[38], the authors make a quantitative bibliometric analysis to evaluate chatbots created in the previous years. Its goal is to help researchers identify research gaps for future chatbot-related work. For this, the authors used four tools, literature databases (WoS and ProQuest), CiteSpace to analyze and group the data and Bibliometrix to determine recurring patterns. The result of the analysis found a lot of research opportunities in chatbots due to the current popularity of Deep Learning, showing that this technology can dictate the direction of research in chatbots. Many other recommendations were obtained from this analysis such as sentiment and behavior analysis.

Table 3 summarizes the survey carried out in the selected articles. The technologies that are the most used in the articles studied are: Frameworks, in three articles, NLP, also in three articles, Sentiment Analysis, in one article, and Timelines, in 1 article.

Until this moment, no studies were found that performed the analysis and comparison of the development of chatbots using the analysis of user sentiments combined with the help of a timeline for the development of the context.

Analyzing the results obtained in this investigation, it appears that most existing chatbots do not have a great performance and tend to harm the image of the company that uses them, instead of improving it. It is also observed that this performance can possibly be improved, by using tree navigation techniques to optimize non-linearity in context and sentiment analysis for better user service.

It was verified the use of two different types of techniques for the development of chatbots. It is known that through tree navigation, it is possible to better direct the dialogue with chatbots, but this technique is little or not used since most existing frameworks only do the basics of NLP, which consists of displaying the probability that the user is chatting with a previously mapped intent. We can also observe that sentiment analysis is rarely used to determine user sentiment or to respond more empathically.

It is also important to note that out of the six studied chatbots only one uses a NLP library to get more information from the dialog, such as text sentiment. Also, three of them use existing chatbot services, like Google’s Dialogflow, and the other two use custom code.

## 4. A New Chatbot Framework

In this section, the proposed chatbot framework is presented.

As presented in Figure 5, the first step of the framework is to customize the chatbot in the *Customization Layer*, building the *Data Object* (chatbot properties, tree timeline structure), and coding the *Custom Functions*. The *Data Object* and the *Custom Functions* are then loaded into the *Framework Layer*, generating a new *Chatbot*. The new *Chatbot* can then send and receive messages from the *User Interface*. Moreover, the structure of the framework and the concepts adopted to build it are discussed and explored.

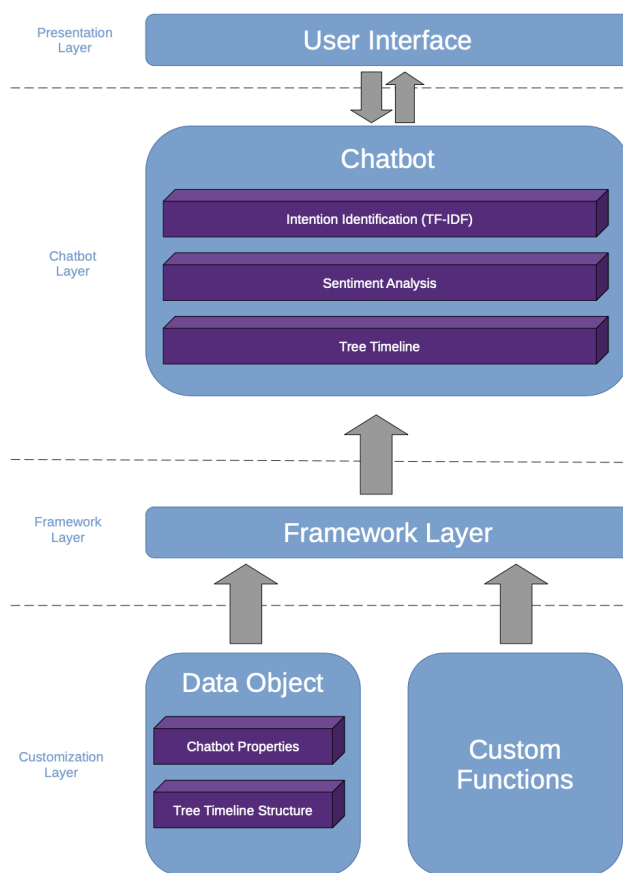


Figure 5. Architecture diagram

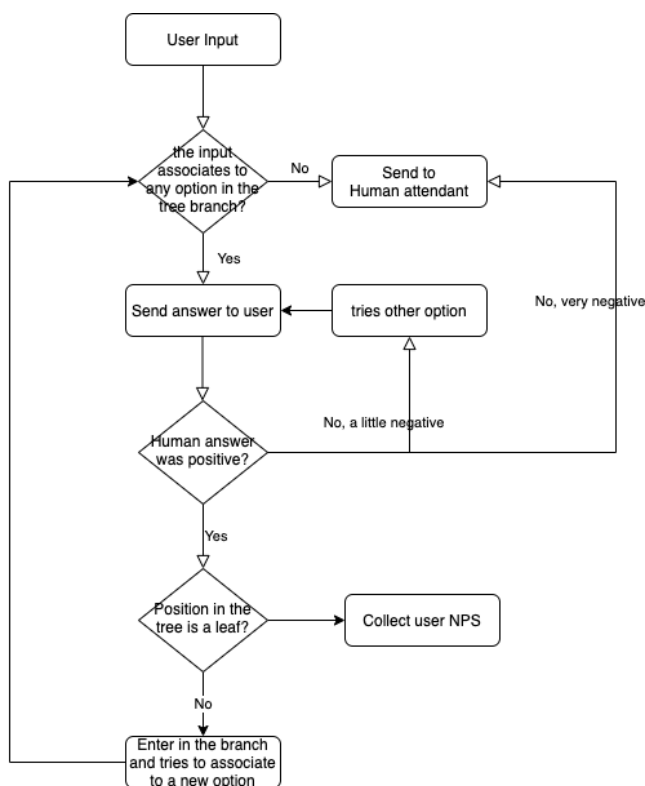
### 4.1 The Framework Structure

As depicted in Figure 6, the proposed framework receives the user input and then searches for an association in the tree. If an association is found, the framework returns an answer, and, otherwise, it redirects it directly to a human attendant. After the message is sent, the framework waits for the user response. If the answer is positive, it checks the next branch of the tree and continues the execution flow. In case there is not another branch in the flow, the framework may ask if the user wants to chat about another topic or finishes the dialogue. Finally, the framework collects feedback from the user.

For example, if the user says “I would like to check my

Title	YEAR	Frame-work	NLP	Custom Code	Sentiment Analysis	Time-Line
Experimentation for Chatbot Usability Evaluation: A Secondary Study	2022	✓	✓	✓	✓	
Chatbot for Healthcare System Using Artificial Intelligence	2020			✓		
Xatkit: A Multimodal Low-Code Chatbot Development Framework	2020			✓		
Automatic Text Summarization of News Articles in Serbian Language	2019		✓			
Convey: Exploring the Use of a Context View for Chatbots	2018	✓				✓
Scalable Sentiment for Sequence-to-Sequence Chatbot Response with Performance Analysis	2018		✓		✓	
A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks	2018	✓				
Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments	2017		✓			
Chatbots and conversational agents: A bibliometric analysis	2017	✓				

**Table 3.** Technologies used in chatbot development of



**Figure 6.** Framework diagram

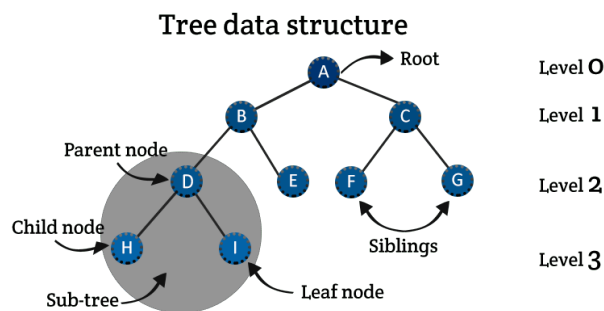
account extract”, the system will search for the node of the tree that has the trained answers that are most similar to the user input, in this example, the “Account Activity” branch and return it to the user. If the user answer with a positive answer,

like “ok, thanks”, the chatbot replies simply by asking if it can help with anything else.

### 4.2 Tree Timeline

In order to create the chatbot dialogue flow, a tree data structure is built. This way, the framework knows the exact flow to achieve its goal.

As illustrated in Figure 7, a tree data structure starts by the top of the tree (root node), and then goes to one of its child nodes. Each dialogue option creates, then, a subtree, with the option flow. In the end of the flow, we can find the last node (leaf node), that finishes this flow.



**Figure 7.** Tree data structure used to represent the timeline

Using the TF-IDF algorithm, the framework can obtain the user intention, and based in this intention it can navigate through the tree, returning to the user the branch that best suits his intention.

This feature provides the chatbot with a conversational flow and prevents it from asking questions that have already



been asked. The key to this feature is the flow itself, which lets the chatbot know the next step for each situation.

For example, if the user asks for the cancellation of the credit card, the chatbot may put this user in the “Credit Card Cancellation” flow and may offer a new credit card with lower taxes or offer more credit.

### 4.3 Sentiment Analysis

The Sentiment Analysis is the feature that provides information to the framework that enables it to infer if the user is having a good or bad experience with the interaction.

Using the VADER Sentiment Library that is built-in the NLTK framework, it’s possible to measure the positivity of the user’s sentence, and understand if the user is happy or not with the given answer

If the user sentiment is positive, it means that the framework is giving the right answers and it is in the right path of the tree timeline. Otherwise, if the sentiment is negative, the chatbot can try another option in the same or another flow (sub-tree). In the case that the sentiment is very negative, it may mean that the user just want to talk with a real person, so the chatbot may redirect the user to a real attendant.

For example, If the answer is quite negative, like “No, I don’t want this”, the system will check other branches of the tree that has a match with the previous user input. Conversely, if the answer is very negative, like “You’re not helping me at all”, the system may then address the user to a human attendant.

## 5. Results and Discussion

In this section, we present results of our experimental study. All experiments presented here are performed considering scenarios in which a fictional bank customer interacts with a developed bank chatbot to solve one or more problems. The chatbot instance has its own timeline tree and custom functions where it can retrieve user data and solve a set of problems. Also, whenever the chatbot fails to solve an issue, it may redirect the user to a human attendant.

### 5.1 Experimental Setup

In our experimental setup, a chatbot is built using the timeline tree depicted in Figure 9. In this case, we simulate a bank chatbot that tries to help users with some requests and, in more complex cases, redirects them to a human attendant.

As shown in figure 8, the chatbot compliments the user and asks for the first input. Based on this input, the chatbot navigates in the timeline tree using intention detection algorithm, providing the user an appropriate answer to his request.

In Figure 8 example, the system search for a match for the sentence “I would like to check my account extract”. It, then, finds a match in the “Account Activity” node and returns its result. After a positive feedback, the system continues the flow of the node or if it is a leaf node, asks if the user needs help with anything else. This is the default conversation flow.

**John Banks:** Hi Mary, welcome to Blue Bank, How can I help you?

**You:** I would like to check my account extract

**John Banks:** Your card activity in the last 7 days: Today: USD: 1000.00

**You:** Ok, thanks

**John Banks:** Can I help you with anything else?

**You:** No, bye

**Figure 8.** Chatbot Example

### 5.2 Sentiment Analysis

Using a sentiment analysis library makes it possible to get the sentiment from the user’s text and then perform actions such as matching with another branch of the tree or sending the user to a human attendant.

In the proposed Framework, we use the NLTK built-in library VADER Sentiment to measure the positivity of the user sentence to ensure the given answer is the right one.

In Figure 10, for example, it can be noticed a high rejection from the user with the chatbot. The chatbot then detects the negative sentiment in the text and redirects the user to a human attendant.

Conversely, in Figure 11, the user informs that the chatbot picked the wrong branch. The chatbot detects the low amount of negativity in the user sentence and then tries another option in the tree.

This decision is made based in the user Positivity in the feedback sentence, by default it’s considered a positive feedback a value higher than -0.05, that is a neutral or positive answer, a feedback between -0.05 and -0.2 would be considered quite negative and a feedback lower than -0.2 is considered very negative. These values can be customized in code and also configured automatically in an online learning strategy.

### 5.3 Tree Timeline

Using a tree timeline makes it possible to create a flux of conversation, like the credit card flux in Figure 9, were the user can choose different options inside a sub-tree, like a new credit card, more credit or cancellation. More complex fluxes can be made, implementing more complex tasks like a more charismatic sign up form or questionnaires.

This feature was implemented using a TF-IDF algorithm to identify the user’s intention and then match it against the chatbot tree. This is the key to avoid the replication of questions that already were made, since once a question was already made and the chatbot already knows the answer, there is no point in making the same question again. This increases chatbots performance and usability.

Without this feature the sentiment analysis and intention identification would also be less efficient, since the dialogues would not have an order, increasing the number of entries that the intention identification would need to search every time. Also, the retry feature using sentiment analysis would decrease the quality. This is also illustrated in Figure 11

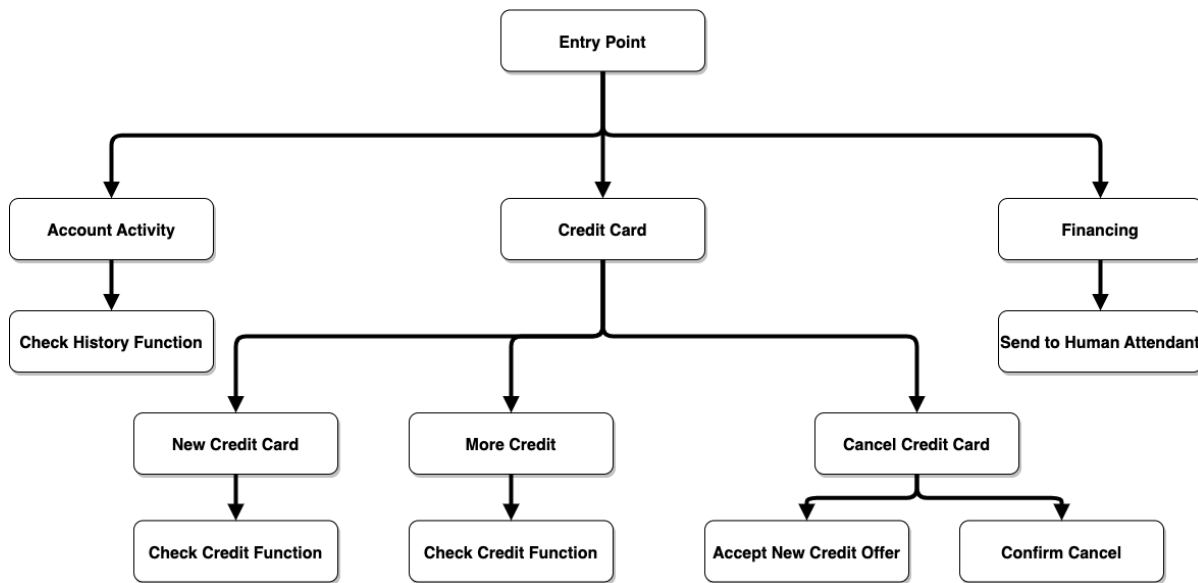


Figure 9. Experiment Tree

**You:** So, I need to upgrade my credit card  
**John Banks:** I see you have a high debt in your credit card, may I provide you with a new line of credit?  
**You:** you're not helping me at all (Sentiment: -0.2235)  
**John Banks:** Sorry, I will redirect you to a human attendant shortly.

Figure 10. Example of detection of negativity in a chatbot dialogue

**John Banks:** Hi Wagner, welcome to Blue Bank, How can I help you?  
**You:** I want to check my credit  
**John Banks:** We are sorry you want to cancel your credit card, we can offer you a limit of USD: 2000, with the tax of USD: 20 per month. Do you want to try our new offer or do you really want to cancel the card?  
**You:** No, I don't want this  
**John Banks:** I see I could not understand your statement. May I provide you with a line of credit? **You:** Yes, please

Figure 11. Example where the chatbot detects the low amount of negative in the sentence

### 5.4 Framework Performance

Based on the proposed experimental setup, the framework was able to accomplish what it was proposed: creating a chatbot with sentiment analysis and tree timelines. In addition to improving user experience, the proposed chatbot is lost less often than a common chatbot and is not repetitive. This happens because once a variable is collected, the system disables the dialog that would request that data, and cases like the ones presented in Figure 12 or Figure 13 are less prone to happen.

The performance of the chatbot may vary accordingly to the Tree Timeline Structure created in the framework.

**John Banks:** Sorry, I didn't understand what you meant.

Figure 12. Example where the chatbot do not understand what the user mean

**You:** What's my account balance?  
**John Banks:** Please insert your Social Security Number:  
**You:** 123-45-6789  
**John Banks:** Your balance is: 1.000,00 USD.  
**John Banks:** How much is my credit card limit?  
**John Banks:** Please insert your Social Security Number:

Figure 13. Example where the chatbot asks multiple times for the same data

## 6. Conclusion

In this work, we present a chatbot-building framework that considers the sentiment and interaction timeline of the user to provide a more precise answer. Using sentiment analysis the chatbot can check the feedback from the user's answer, and by using a timeline structure, it can avoid repeating the same previous mistakes as most common chatbots tend to make.

In our experiments, we show that these new features make difference in the chatbot development and create more developer options for chatbot answer customization, and even create better checkpoints to redirect users to a human attendant.

As future work, we recommend an optimization of the framework usability, as present in most recent works, by creating a web interface where a non-developer can create, train, and verify the usability of a chatbot, without having to write code in a specific language. This would lead to the increase in the amount of users and generated chatbots. We also intend to make a qualitative evaluation of the generated chatbots,

comparing them in order to assess user acceptance of the new built-in features.

## Author contributions

All authors have contributed equally to the development of this work.

## References

- [1] BAO, Q.; NI, L.; LIU, J. HHH: An online medical chatbot system based on knowledge graph and hierarchical bi-directional attention. In: *ACSW '20: Proceedings of the Australasian Computer Science Week Multiconference*. New York, USA: Association for Computing Machinery, 2020. p. 1–10. Disponível em: <<https://doi.org/10.1145/3373017.3373049>>. Acesso em: 05 de julho de 2020.
- [2] WEIZENBAUM, J. ELIZA — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, New York, USA, v. 9, n. 1, p. 36–45, January 1966.
- [3] SHAWAR, B. A.; ATWELL, E. Different measurements metrics to evaluate a chatbot system. In: *Proceedings of the NAACL'07 Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*. New York, USA: Association for Computational Linguistics, 2007. p. 89–96.
- [4] TAYLOR, A. S. Machine intelligence. In: *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, USA: Association for Computing Machinery, 2009. p. 2109–2118. Disponível em: <<https://doi.org/10.1145/1518701.1519022>>. Acesso em: 05 de julho de 2020.
- [5] GRAY, J. What next? a dozen information-technology research goals. *Journal of the ACM*, Association for Computing Machinery, New York, USA, v. 50, n. 1, p. 41–57, January 2003. Disponível em: <<https://doi.org/10.1145/602382.602401>>. Acesso em: 05 de julho de 2020.
- [6] NURUZZAMAN, M.; HUSSAIN, O. K. A survey on chatbot implementation in customer service industry through deep neural networks. In: *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. New York, USA: IEEE, 2018. p. 54–61.
- [7] NIMAVAT, K.; CHAMPANERIA, T. Chatbots: An overview. types, architecture, tools and future possibilities. *International Journal for Scientific Research & Development*, India, v. 5, n. 7, p. 1019–1024, October 2017. Disponível em: <[https://www.researchgate.net/publication/320307269\\_Chatbots\\_An\\_overview\\_Types\\_Architecture\\_Tools\\_and\\_Future\\_Possibilities](https://www.researchgate.net/publication/320307269_Chatbots_An_overview_Types_Architecture_Tools_and_Future_Possibilities)>. Acesso em: 22 de junho de 2020.
- [8] MÜLLER, L. et al. Chatbot acceptance: A latent profile analysis on individuals' trust in conversational agents. In: *SIGMIS-CPR '19: Proceedings of the 2019 on Computers and People Research Conference*. New York, USA: ACM, 2019. p. 35–42.
- [9] CHEN, H. et al. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, Association for Computing Machinery, New York, USA, v. 19, n. 2, p. 25–35, December 2017. Disponível em: <<https://doi.org/10.1145/3166054.3166058>>. Acesso em: 22 de junho de 2020.
- [10] KLOPFENSTEIN, L. C. et al. The rise of bots: A survey of conversational interfaces, patterns, and paradigms. In: *DIS '17: Proceedings of the 2017 Conference on Designing Interactive Systems*. New York, USA: Association for Computing Machinery, 2017. p. 555–565. Disponível em: <<https://doi.org/10.1145/3064663.3064672>>. Acesso em: 22 de junho de 2020.
- [11] LEE, C.-W. et al. Scalable sentiment for sequence-to-sequence chatbot response with performance analysis. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New York, USA: IEEE, 2018. p. 6164–6168.
- [12] RIEHLE, D. *Framework design: A role modeling approach*. Tese (Doutorado) — ETH Zurich, Zürich, Switzerland, 2000.
- [13] LEVINE, J. R. *Linkers & loaders*. Burlington, Massachusetts, USA: Morgan Kaufmann Publishers, 1999.
- [14] SAUVÉ, J. *O que é um framework?* 2000. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>>. Acesso em: 3 de janeiro de 2023.
- [15] MAROENGSI, W. et al. A survey on evaluation methods for chatbots. In: *ICIET 2019: Proceedings of the 2019 7th International Conference on Information and Education Technology*. New York, USA: Association for Computing Machinery, 2019. p. 111–119.
- [16] NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, Oxford University Press, London, United Kingdom, v. 18, n. 5, p. 544–551, September 2011.
- [17] DALE, R. *Automated Writing Assistance: Grammar Checking and Beyond Topic 5: Beyond the Sentence*. 2011. Disponível em: <<https://www.slideshare.net/rdale01/tarragona-summer-schoolautomated-writing-assistance-topic-5>>. Acesso em: 3 de janeiro de 2023.
- [18] MACHEREY, K.; OCH, F. J.; NEY, H. Natural language understanding using statistical machine translation. In: *7th European Conference on Speech Communication and Technology (Eurospeech 2001)*. Netherlands: ISCA, 2001. p. 2205–2208. Disponível em: <[https://www.isca-speech.org/archive\\_v0/archive\\_papers/eurospeech\\_2001/e01\\_2205.pdf](https://www.isca-speech.org/archive_v0/archive_papers/eurospeech_2001/e01_2205.pdf)>. Acesso em: 10 de dezembro de 2019.

- [19] YI, J. et al. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: *Third IEEE international conference on data mining*. New York, USA: IEEE, 2003. p. 427–434.
- [20] NASUKAWA, T.; YI, J. Sentiment analysis: Capturing favorability using natural language processing. In: *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*. New York, USA: ACM, 2003. p. 70–77.
- [21] HUTTO, C.; GILBERT, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAI Conference on Web and Social Media*, AAAI Press, Palo Alto, USA, v. 8, n. 1, p. 216–225, May 2014.
- [22] OMRAN, F. N. A. A.; TREUDE, C. Choosing an nlp library for analyzing software documentation: a systematic literature review and a series of experiments. In: *Proceedings of the 14th International Conference on Mining Software Repositories*. New York, USA: IEEE, 2017. p. 187–197.
- [23] KOSMAJAC, D.; KEŠELJ, V. Automatic text summarization of news articles in serbian language. In: *18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. New York, USA: IEEE, 2019. p. 1–6.
- [24] LOPER, E.; BIRD, S. Nltk: the natural language toolkit. *ETMTNLP '02: Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, New York, USA, v. 1, p. 63–70, July 2002. Disponível em: <<https://arxiv.org/abs/cs/0205028>>. Acesso em: 10 de dezembro de 2019.
- [25] HONNIBAL, M.; MONTANI, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. No prelo. 2017.
- [26] AL-RFOU, R.; PEROZZI, B.; SKIENA, S. Polyglot: Distributed word representations for multilingual nlp. In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. USA: Association for Computational Linguistics, 2013. v. 1307.1662, p. 183–192. Disponível em: <<https://arxiv.org/abs/1307.1662>>. Acesso em: 10 de dezembro de 2019.
- [27] ALBERTI, C. et al. Syntaxnet models for the conll 2017 shared task. *arXiv preprint arXiv:1703.04929*, p. 1–6, March 2017. Disponível em: <<https://arxiv.org/abs/1703.04929>>. Acesso em: 10 de dezembro de 2019.
- [28] MANNING, C. et al. The Stanford CoreNLP natural language processing toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014. p. 55–60. Disponível em: <<https://www.aclweb.org/anthology/P14-5010>>. Acesso em: 10 de dezembro de 2019.
- [29] BRADESKO, L.; MLADENIC, D. A survey of chatbot systems through a loebner prize competition. In: *Proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies*. [S. l.]: [s.n.], 2012. p. 34–37.
- [30] JAIN, M. et al. Convey: Exploring the use of a context view for chatbots. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, USA: ACM, 2018. p. 1–6.
- [31] MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. Cambridge, United Kingdom: Cambridge University Press, 2008.
- [32] KARUMANCHI, N. *Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles*. 5. ed. Hyderabad, India: CareerMonk Publications, 2016.
- [33] LIAO, Q. V. et al. All work and no play? In: *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, USA: Association for Computing Machinery, 2018. p. 1–13. Disponível em: <<https://doi.org/10.1145/3173574.3173577>>. Acesso em: 22 de junho de 2020.
- [34] HU, T. et al. Touch your heart: A tone-aware chatbot for customer care on social media. In: *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, USA: ACM, 2018. p. 1–12.
- [35] REN, R. et al. Experimentation for chatbot usability evaluation: A secondary study. *IEEE Access*, IEEE, New York, USA, v. 10, p. 12430–12464, 2022.
- [36] ATHOTA, L. et al. Chatbot for healthcare system using artificial intelligence. In: *8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. New York, USA: IEEE, 2020. p. 619–622.
- [37] DANIEL, G. et al. Xatkit: A multimodal low-code chatbot development framework. *IEEE Access*, IEEE, New York, USA, v. 8, p. 15332–15346, January 2020.
- [38] IO, H.; LEE, C. Chatbots and conversational agents: A bibliometric analysis. In: *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. New York, USA: IEEE, 2017. p. 215–219.